Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

AV - Ambiente Virtual

CSCW - Computer Supported Cooperative Work

EAD - Educação a Distância

EAI - External Authoring Interface

EVI - Educação Virtual Interativa

HMD - Head Mounted Display

HTML - Hyper Text Markup Language

JVM- Java Virtual Machine

LVF - Laboratório Virtual de Física

MEC - Ministério da Educação e Cultura

PSPICE - Simulations Program with Integrated Circuit Emphasis

RV - Realidade Virtual

VRML - Virtual Reality Modeling Language

Lista de Publicações Relacionadas a este Trabalho:

BORGES E.; CARDOSO A.; LAMOUNIER E. O Uso de Realidade Virtual na Visualização e Corte de Poliedros. In:4TH SYMPOSIUM ON VIRTUAL REALITY - **SVR 2001** - **Resumos.** Florianópolis, SC, Oct. 2001, p. 280-290.

CARDOSO, A.; LAMOUNIER E.; TORI R. Sistema de Criação de Experiências de Física em Realidade Virtual para Educação a Distância. In: II WORKSHOP BRASILEIRO DE REALIDADE VIRTUAL. **WRV´99**, Marília , São Paulo, SP, 1999, p. 174-181.

CARDOSO, A.; LAMOUNIER E.; TORI R. A Virtual Physics Laboratory to Support Interactive Distance Learning In: INTERNATIONAL CONFERENCE ON ENGINEERING AND COMPUTER EDUCATION - **ICECE´2000**. São Paulo, S.P., ago. 2000, 1 CDRom.

CARDOSO, A.; LAMOUNIER E.; TORI R. Realidade Virtual no Ensino de Física a Distância e Presencial In: XI SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO - **SBIE´2000.** Maceió, Alagoas, nov. 2000- p. 476-478.

CARDOSO A.; LAMOUNIER E.; TORI R. Virtual Physics Experiments to Support Interactive Distance Learning In: COMPUTER GRAPHICS AND IMAGING - **CGIM 2001** - IASTED International Conference , Honolulu - Hawaii - USA, ago. 2001, v. 01, n. 01, p. 07 -12.

CARDOSO A.; LAMOUNIER E.; TORI R. Interactive 3D Physics Experiments Through the Internet. In: 4TH SYMPOSIUM ON VIRTUAL REALITY- **SVR 2001**. Florianópolis, SC, Oct. 2001, p. 280-290.

CARRIJO .R.; CARDOSO A.; LAMOUNIER E. Um Algoritmo Genérico Aplicado ao Controle de Humanóides em Ambientes Virtuais Interativos. In: 4TH SYMPOSIUM ON VIRTUAL REALITY- **SVR 2001**. Florianópolis, SC, Oct. 2001, p. 211-216.

FARIA L. et al. A System to Evaluate Wasted Electric Energy Based on Virtual Reality Techniques. In: COMPUTER GRAPHICS AND IMAGING - **CGIM 2002** - IASTED International Conference, Kauai, Hawaii, USA. Ago. 2002.

FARIA L. et al. Desenvolvimento de Técnicas de Realidade Virtual para o Ensino/Conscientização do Consumo de Energia Elétrica. In: Symposium on Virtual Reality 2002 - **SVR 2002**, Fortaleza, CE, oct. 2002, submetido.

1. Introdução

1.1. Preâmbulo

Tecnologias como Hipermídia e Realidade Virtual (RV), oferecem um grande potencial para a evolução tanto da educação presencial quanto da educação a distância, pelo fato de possibilitarem o aumento da interatividade, abrindo espaço para a chamada Educação Virtual Interativa (EVI) (Tori; Ferreira, 1999). A análise do panorama atual de aplicação das tecnologias interativas na Educação, permite as seguintes constatações:

i) a Internet tem sido amplamente utilizada nos mais diversos segmentos de mercado (Grove, 1999), inclusive no segmento educacional, de tal forma que autores como Hiltz; Wellman (1997) apud Slouka´s (1995) afirmam que a comunicação face a face vem perdendo espaço para a comunicação mediada por computador;

ii) há proposições de soluções inovadoras, do ponto de vista educacional, visando o aprimoramento de métodos de ensino/aprendizagem, seja com utilização de Hipermídia (Dobson; Riesbeck, 1998) ou Realidade Virtual (Byrne, 1996); (Keller; Schreiber, 1999); (Carswell,1997); (Kiril et al., 2000), compatíveis com a Internet;

iii) há disponibilidade de tecnologias relacionadas à Realidade Virtual, capazes de proverem condições de desenvolvimento de laboratórios virtuais educacionais;

iv) a utilização de RV com fins educacionais tem mostrado bons resultados relativos à aprendizagem (Byrne, 1996); (Pausch et al., 1997);

v) a Educação a Distância (EAD), hoje largamente aceita pela sociedade e até incentivada pelo Governo Brasileiro (MEC, 2001), é uma importante forma de ensino, que tem ganhado muita força com o uso das telecomunicações, do computador e, mais recentemente, da Internet.

Apesar deste ambiente propício e do grande potencial da EVI, não se verifica ainda uma disseminação de aplicações de Realidade Virtual na Educação no mesmo ritmo percebido em tecnologias baseadas apenas em Hipermídia via *World Wide Web* (Web).

Analisando aspectos relativos aos resultados providos pela utilização de tais tecnologias, Schank (2001) aponta que a simples substituição de aulas presenciais por apostilas Hipermídia ou vídeos, através da Internet, é caracterizada por perdas significativas no processo de aprendizado, principalmente pelo fato de o aluno não poder "aprender fazendo". Diante de tais constatações, verifica-se que uma das aplicações potenciais de RV na educação está relacionada com o desenvolvimento de laboratórios virtuais, que, apesar de não serem comparáveis a um verdadeiro laboratório, possibilitam aos aprendizes sensações mais próximas das relativas à experiência real.

O desenvolvimento destes laboratórios, compatíveis com a Web, constitui-se em importante recurso a situações em que barreiras geográficas ou de custos inviabilizam o acesso a laboratórios reais, ou mesmo como apoio didático complementar a aulas teóricas, seja para cursos a distância, seja para cursos presenciais.

A análise dos produtos e soluções relativos a laboratórios virtuais educacionais, permite constatar as principais limitações a suas utilizações, seja pela exigência de conhecimento de alguma forma de programação para o uso satisfatório destes ambientes (Green; Halliday, 1996), seja por barreiras relativas ao custo de montagem e implementação.

Neste sentido, esta tese faz um levantamento das ferramentas e soluções existentes para o desenvolvimento e utilização de laboratórios virtuais (2D e 3D) suportados pela Internet e, dentro da ampla possibilidade de pesquisa na área de Realidade Virtual (RV) aplicada à Educação, analisa os pontos fortes e fracos de tais soluções, de forma a identificar as limitações tecnológicas que dificultam a disseminação desta tecnologia como ferramenta educacional, e propõe soluções que contribuam para minimizar tais limitações.

Para os propósitos deste trabalho, optou-se por delimitar o estudo em um domínio específico do conhecimento, a Física Clássica para alunos do ensino médio, e, com base nos resultados da análise das ferramentas disponíveis, propõe-se uma arquitetura para uma solução tecnológica que possibilite o desenvolvimento e configuração de ambientes virtuais

interativos por professores, sem exigência de conhecimentos específicos de técnicas de programação e Computação Gráfica.

1.2. Motivação

Segundo Myers (1998), pesquisas nas universidades e/ou corporações devem investigar novas alternativas de interface no sentido de melhoria do processo interativo, principalmente na Web. Neste processo investigativo, destaca-se como área emergente a concepção de espaços de navegação tridimensionais (3D) e Realidade Virtual (RV) .

Com o propósito de concepção de software para RV, investigam-se técnicas de modelagem 3D (Kirner, 1997); (Byron et al., 1996) e o desenvolvimento de aplicativos para a Internet nos últimos anos (Pausch et al., 1997); (Shermam et al., 1997).

Uma das áreas beneficiadas por essas pesquisas é a educacional, onde há enorme potencial para desenvolvimentos de pesquisas e projetos com uso de Realidade Virtual, que permite aos usuários experimentarem sensações únicas através da manipulação e análise do objeto de estudo (Pinho, 2000b).

Analisando o meio de entrega, observa-se que soluções para a Internet estão associadas à fácil acessibilidade e à independência de plataforma (Basal, 1999) e que o uso de laboratórios virtuais como suporte à interação pode ser mais eficiente, conveniente e até mais barato do que métodos tradicionais (Cardoso et al., 2000a), já que ferramentas gratuitas podem ser usadas nas suas criações e no desenvolvimento. Tais fatores têm motivado o desenvolvimento de diversas proposições de ambientes virtuais 3D educacionais (Byrne, 1996); (Costa, 2000); (Ipolito, 2000); (Keller, 1999); (Lawton, 1999); (Pantelidis, 1996).

No entanto, observa-se que a maioria dessas soluções, como as apresentadas por Chen et al (1997), Kiril et al. (2000), Meiguins; Behrens (1999), Ost et al. (1999), Reddy et al. (1999), Rudomim (1999); Silva; Osório (2000) não permitem que professores desenvolvam seus próprios experimentos e lições, limitando-se a permitir a utilização de soluções envolvendo mundos virtuais pré-concebidos .

Tais limitações obstruem a flexibilidade de desenvolvimento criativo. Outras soluções, como as propostas de Martins et al. (1999) e Pierce (1997), se caracterizam por ambientes de fácil interação e que permitem criação de novos mundos virtuais, mas têm

limitações de aplicabilidade, seja por se basearem em tecnologias não gratuitas, seja pela necessidade do usuário de aprender linguagens específicas para usar o software. Isto é um problema, uma vez que a grande maioria de educadores que têm acesso à Internet não possuem habilidades técnicas e computacionais que lhes permitam conceber seus próprios mundos virtuais.

Atualmente, um professor que deseja utilizar, didaticamente, recursos de RV em ensino, dispõe, em geral, das seguintes opções:

(a) Uso de produtos na forma de aplicativos que ofereçam experimentos pré-desenvolvidos;

(b) Uso de produtos disponíveis em bibliotecas de componentes, constituídos por aplicações para navegadores da Internet (como *applets* Java ou ambientes virtuais desenvolvidos em VRML), em alguns endereços da Internet (Labvirt, 2000); (Terra, 2000); (Kiril et al., 2000); (Physicsweb, 2000);

(c) Processo de desenvolvimento próprio de experimentos virtuais, o que implica em aprender a desenvolver programas ou conceber lições usando ferramentas genéricas disponíveis na Internet, como o *Alice* (Pierce et al., 1997), linguagens para tais fins como o VRML (Ames et al., 1997); (Mark, 1999) ou *Java* (Chan et al., 1999); (Ritchey, 1997);

(d) Desenvolvimento terceirizado, por meio de encomenda de produto(s) desejado(s) a especialistas ou empresas geradoras de programas computacionais;

(e) Desenvolvimento na própria instituição, por meio de uma equipe de especialistas designada especificamente para este fim.

A análise das possibilidades, acima apresentadas, permite identificar que as soluções (a), (d) e (e) são, via de regra, dispendiosas, limitando a quantidade de usuários que a ela têm acesso.

A opção (b) tem como inconveniente a impossibilidade de modificação dos conteúdos encontrados, sendo necessário utilizar as soluções pré-definidas, que nem sempre estão em sintonia com a forma didática desejada pelo professor em questão.

O aprendizado de ferramentas genéricas - item (c), gera grande demanda de tempo e a necessidade de aprendizado de requisitos básicos de técnicas computacionais, desviando o professor de sua atividade fim.

Quando o desenvolvimento é relacionado à criação de um sistema RV, agravam-se itens como a necessidade de domínio de modelagem geométrica tridimensional, simulação em tempo real, navegação, colisão, projeto de interfaces e aplicações simples e distribuídas em diversas áreas (Kirner, 1997); (Martins et al., 1999).

Green; Halliday (1996) apontam que uma das principais necessidades de pesquisa nesta área está relacionada com a redução do esforço requerido para a criação de novos ambientes virtuais, já que, geralmente, há a exigência de alguma forma de programação. Este fator aumenta o custo de desenvolvimento, uma vez que equipes de produção necessitam incluir um programador (nem sempre familiar ao domínio da aplicação). Uma forma de redução do custo de construção é a concepção de novas ferramentas, acessíveis tanto a programadores quanto a não-programadores.

Por outro lado, discutindo aplicações de RV, Johnson (1998) aponta que ambientes virtuais suportados por monitores convencionais criam uma grande oportunidade para concepção de interfaces inovadoras e que técnicas tradicionais de interação falham no suporte a esta forma de desenvolvimento. Para evitar tais falhas é necessário centrar o desenvolvimento na diversidade de tarefas diárias da população envolvida, identificando-as de forma que a análise não seja constituída de suposições sobre as tarefas envolvidas com o público alvo e voltar o desenvolvimento para um dado domínio (forte definição de um foco), de forma a evitar falhas graves na análise de tarefas do sistema.

Em face do exposto, justificam-se pesquisas no sentido de geração de sistemas para a Internet, voltadas para um dado domínio, que permitam a professores a utilização de técnicas de Realidade Virtual como suporte a Educação, em particular por meio de laboratórios virtuais. É importante destacar que estas técnicas devem permitir aos educadores desenvolverem seus próprios mundos virtuais, motivando-os a experimentar, sem barreiras de longo tempo de aprendizado ou de serem obrigados a utilizar soluções didáticas pré-definidas.

O foco deste trabalho está no desenvolvimento de uma plataforma computacional voltada a um domínio de aplicação que, utilizando Realidade Virtual, permita a professores, desprovidos de conhecimento sobre técnicas de programação de computadores, desenvolver lições e experimentos virtuais que possam ser utilizadas em aulas tradicionais ou a distância.

1.3. Objetivos

O objetivo central deste trabalho é, pois, propor e demonstrar a viabilidade de uma arquitetura de sistema que suporte a elaboração de experimentos virtuais interativos de Física em ambientes tridimensionais, que possa ser usado inclusive por professores desprovidos de conhecimento de técnicas de programação e Computação Gráfica. Para tanto, propõe-se a:

(a) Analisar, destacando suas principais vantagens e desvantagens, programas e técnicas computacionais que utilizem Realidade Virtual como ferramenta de ensino e aprendizagem;

(b) Especificar a arquitetura de um sistema computacional caracterizado por interfaces amigáveis, suportado por técnicas de metáforas espaciais (Tori, 1996), que possibilite a professores não iniciados em tópicos de programação em Computação Gráfica ou RV criarem, modificarem e utilizarem mundos virtuais interativos como ferramentas para o ensino e aprendizagem de Física;

(c) Implementar protótipos experimentais que demonstrem a viabilidade da arquitetura proposta, com o uso de linguagens e ferramentas de domínio público;

(d) Avaliar o sistema desenvolvido (protótipos) junto a professores e alunos do ensino médio, de forma a realimentar o processo de análise e projeto.

1.4. Contribuições

Este trabalho apresenta como principais contribuições:

(a) Análise comparativa de sistemas disponíveis relacionados com laboratórios virtuais educacionais;

(b) Definição da arquitetura, identificação dos passos e análise de riscos, para um sistema inovador de criação de ambientes virtuais associados a experiências interativas, relacionadas a um domínio do conhecimento, capaz de ser utilizado por usuários desprovidos de prévio conhecimento em Computação Gráfica e programação;

(c) Desenvolvimento de protótipos de validação do exposto em (b);

(d) Análise de resultados, a partir da submissão dos protótipos a usuários convencionais;

(e) Identificação das ações, visando a continuidade do desenvolvimento, para adequação e aprimorando da arquitetura proposta.

1.5. Organização da Tese

Este trabalho está dividido em sete capítulos, incluindo esta introdução e está assim organizado:

Para o entendimento das barreiras relacionadas com as interfaces convencionais e as soluções existentes, o capítulo dois apresenta a Realidade Virtual, destacando sua importância com respeito à educação, interação homem-máquina e problemas relacionados com interfaces e Educação Virtual Interativa, que são os principais fundamentos do trabalho desenvolvido.

O capítulo três apresenta sistemas relacionados com Educação Virtual Interativa, distinguindo alguns sistemas e soluções disponíveis, apresentando suas principais limitações e contribuições.

O projeto do sistema proposto está ilustrado no capítulo quatro em três tópicos principais: especificação do sistema, arquitetura do sistema e das interfaces. Detalhes do projeto são relatados e apresentados de forma a permitir o processo de desenvolvimento, além dos requisitos e das fases de análise do software desenvolvido.

O capítulo cinco retrata a implementação de alguns protótipos de partes do sistema, apresentando as ferramentas utilizadas, os requisitos necessários, detalhes de códigos, alguns desafios envolvidos com a implementação e com o desenvolvimento das interfaces, além de decisões de projetos.

O capítulo seis apresenta resultados e limitações, além da avaliação dos protótipos pelos usuários, apontando problemas relatados, sugestões, resultados de utilização e as limitações dos protótipos.

As conclusões e sugestões para trabalhos futuros são apresentadas no último capítulo desta tese.

Por fim, a lista de referências - bibliografia - é apresentada. Compõem este trabalho dois anexos: o primeiro apresenta o diagrama de classes relativo à arquitetura proposta no capítulo quatro e o segundo anexo, o questionário encaminhado a usuários do protótipo de forma a identificar sugestões e aplicabilidade.

2. Educação e Realidade Virtual

2.1. Interação na Educação

2.1.1. Educação a Distância e Educação Presencial

"Com o avanço tecnológico o ensino a distância hoje é visto não só como um sistema especial, mas como uma parte integrante do aprendizado que prepara profissionais nas diversas áreas de ensino e pesquisa. Muitas instituições educacionais estão respondendo a estes desafios, aperfeiçoando e desenvolvendo programas de Educação a Distância (EAD). Recursos como redes de trabalho, videoconferência, computação colaborativa, Internet, e outros, estão sendo utilizados para uma maior interação do aluno com o professor, melhorando a qualidade do ensino e difundindo educação em diferentes locais e regiões do país e do mundo" (Sanches; Padro, 2000).

Educação a Distância - EAD, há muito é considerado como uma forma de aprendizado capaz de preparar profissionais em diversas áreas de ensino e pesquisa. Niskier (1999) conclui que EAD torna-se uma possibilidade (instrumento) de qualificação de processo pedagógico e do sistema educacional como um todo, ao contribuir, de forma significativa, para o resgate de valores pedagógicos e propiciar o exercício pleno da cidadania.

A Tabela 2.1 promove uma comparação entre os sistemas de ensino presencial e a distância (Aretio, 1994).

Tabela 2.1 - Comparação entre Ensino Presencial e a Distância

PRESENCIAL	A DISTÂNCIA
ALUNOS	
Homogêneos quanto à idade	Heterogêneos quanto à idade
Homogêneos quanto à qualificação	Heterogêneos quanto à qualificação
Homogêneos quanto ao nível de escolaridade	Heterogêneos quanto ao nível de escolaridade
Lugar único de encontro	Estudam em casa, local de trabalho, etc...
Residência local	População dispersa
Situação controlada / Aprendizagem dependente	Situação livre / Aprendizagem independente
A maioria não trabalha. Habitualmente crianças / adolescentes / jovens	A maioria é adulta e trabalha.
Realiza-se maior interação social.	Realiza-se menor interação social.
A educação é atividade primária. Tempo integral.	A educação é atividade secundária. Tempo parcial.
Seguem, geralmente, um currículo obrigatório.	O próprio estudante determina o currículo a ser seguido.
DOCENTES	
Um só tipo de docente	Vários tipos de docentes
Fonte de conhecimento	Suporte e orientação da aprendizagem
Recurso insubstituível	Recurso substituível parcialmente
Juiz supremo da atuação do aluno	Guia de atualização do aluno
Basicamente: educador	Basicamente: produtor de material ou tutor
Suas habilidades e competências são muito difundidas	Suas habilidades e competências são menos conhecidas
Problemas normais em *design*, desenvolvimento e avaliação curricular	Problemas para o *design*, o desenvolvimento e a avaliação curricular relativos à formação especializado do professor
Os problemas anteriores dependem do professor	Os problemas anteriores dependem do sistema
COMUNICAÇÃO / RECURSOS	
Ensino face a face	Ensino multimídia
Comunicação direta	Comunicação diferenciada em espaço e tempo
Oficinas e laboratórios próprios	Oficinas e laboratórios de outras instituições
Uso limitado de meios	Uso massivo de meios

A análise da tabela permite identificar que EAD provê condições de aprendizado assíncrono, dispensado o formalismo e a necessidade de encontros constantes entre aprendiz e professor. Em um país da dimensão do Brasil, esta característica é especialmente importante.

Uma das principais críticas ao ensino a distância está exatamente na fraca interação da comunidade envolvida. Com respeito ao papel do professor, percebe-se que o mesmo

passa a ser um guia para o aprendiz ao invés de um juiz, fazendo o papel de um tutor ao invés de um educador, na concepção da palavra

Do ponto de vista dos recursos e da comunicação, a análise da Tabela permite constatar que a comunicação utilizada para EAD está fortemente centrada em utilização de multimídia, podendo ser efetuada com separação em espaço e tempo entre professor e aprendiz. Por esta razão, há possibilidade (e até necessidade) de diversos professores estarem envolvidos na produção do material a ser utilizado em EAD, já que não é necessário ao professor detentor do conhecimento relativo ao assunto, o conhecimento de tecnologias relativas a concepção de tais lições com uso de computadores.

Observa-se ainda que na EAD há grande importância no sistema, que é fator fundamental para a obtenção de bons resultados. Assim, o investimento na concepção do sistema é fator de preocupação e geralmente envolve equipes multidisciplinares.

Relativamente à concepção da parte prática, o uso de laboratórios virtuais é uma das grandes vantagens do ensino a distância, de forma que um dado aprendiz pode utilizar recursos de outras instituições que estão disponibilizados remotamente, de forma interativa, como se estivesse dentro de um laboratório real.

Considerando a evolução da Educação a Distância, Saba (1999) relata que EAD passa por três fases cronológicas, relativas aos meios de comunicação (meios de interação) utilizados, a saber:

i) primeira fase (predominantemente até a década de 1960): EAD correspondente à geração textual, com suporte e apoio pedagógicos fundamentados em textos instrucionais impressos;

ii) segunda fase (predominantemente entre as décadas de 1960-1980): EAD composta por textos impressos complementados com recursos tecnológicos de multimídia, tais como gravação de áudio e vídeo;

iii) terceira fase (atual): EAD caracterizada pelo auto-aprendizado com suporte de recursos tecnológicos altamente diferenciados, como a integração das redes de computadores e computadores pessoais, provendo formas de comunicação síncrona e assíncrona .

O grande diferencial desta última fase está nas possibilidades de estabelecimento da comunicação síncrona, tida como capaz de prover maiores condições de reflexão e de elaboração de novos saberes pelos alunos (Aragão et al., 2000).

Sob este ponto de vista, tecnologias interativas (como Multimídia, Hipermídia e Realidade Virtual) são apontadas como propulsoras de uma nova realidade, ao serem empregadas como suporte de ensino e aprendizagem na Educação a Distância (EAD), via Internet. Neste tempo, esta é a verdadeira inovação, propiciada por tecnologias capazes de prover alta interatividade, que, aliadas a redes de computadores podem ser ainda mais extensíveis. Considerando este contexto, faz-se necessária uma análise da terminologia empregada.

O termo EAD, principalmente quando relacionado à terceira fase, torna-se ponto de discussão. Distância, associada ao termo, ao invés de tratar o efeito do ponto de vista do usuário, que se refere à aproximação, acaba por dar uma idéia de separação e obstáculo. Diante disto, Tori; Ferreira (1999) propõem uma adequação da terminologia, associada ao conceito de Educação Virtual Interativa - EVI. EVI é aplicada de forma a representar meios de aprendizagem que eliminam distâncias (não apenas espaciais, mas também temporais e operacionais) e possibilitam catalisar eficiência na aprendizagem, usando tecnologias interativas.

2.1.2. Educação Virtual Interativa - EVI

Considerando os meios relativos a EVI, a comunicação mediada por computador suportada pela Internet é capaz de habilitar pessoas com interesses comuns, separadas por distâncias físicas, a estabelecerem e manterem profícuas relações, definindo verdadeiras comunidades virtuais (Hiltz; Wellman, 1997).

Considerando os processos que suportam tais relacionamentos (síncronos e assíncronos) Raposo et al. (1999) ressaltam alguns problemas vivenciados por tais comunidades, relacionados com as barreiras significativas com respeito à colaboração na WEB apresenta. Um destes problemas refere-se à não-homogeneidade de tais comunidades. De posse destas constatações, faz-se necessária uma discussão relacionada ao meio e à forma de aplicá-lo, principalmente, como ferramenta educacional.

De forma a identificar meios, Lawhead et al. (1997) apresentam as razões que tornaram a Internet uma ferramenta atrativa para EVI:

a) Acessibilidade: ela está acessível a diferentes usuários (professores e alunos), propiciando cursos em áreas remotas, instrução em qualquer horário e pesquisa computacional;
b) Independência de plataformas, associada a grande diversidade de sistemas operacionais e hardware existentes;
c) Transparente "mudança em modalidade": pode-se usar diversas modalidades, sem a mudança do equipamento;
d) Globalização dos cursos: possibilidade de internacionalização de cursos, difundindo rapidamente novos conceitos;
e) Ênfase em criação de aprendizes como produtores: os aprendizes podem participar de forma ativa do processo, tornando-se desenvolvedores.

Baseados na utilização da Internet e de forma a estabelecer parâmetros de diferenciação, Tori; Ferreira (1999) distinguem três formas básicas de EVI:

I) EVI presencial: caracterizada pelo deslocamento dos estudantes até um centro - "Centro de EVI", onde participam de aulas presenciais, nesta modalidade, a tecnologia se presta a diminuir apenas a distância operacional;
II) EVI remota: o estudante define o local de seu aprendizado e pode participar de uma EVI síncrona, em que as informações são trocadas simultaneamente entre professores e alunos, como uma vídeo-conferência ou assíncrona, em que a troca de informações se dá, por exemplo, por e-mail;
III) EVI mista: compreende uma união das duas formas anteriores, com reuniões periódicas do grupos e o estudo remoto.

A terceira forma - EVI mista - tem como exemplo de aplicação a disciplina de cursos de pós-graduação *Hipermídia e Multimídia* da Universidade de São Paulo. Neste caso, recursos de lista de discussão, chat e auto-aprendizagem fazem parte da rotina do professor e dos alunos desde 1995 (Tori; Ferreira, 1999).

Nesta proposta, algumas aulas são realizadas em laboratório com terminais conectados a Internet e a pesquisa de seminários e monografias, como parte da avaliação da disciplina, vêm sendo desenvolvidas via Internet. Os resultados são divulgados

continuamente nas páginas da disciplina, de forma a motivar os grupos de alunos à interação. A própria página principal da disciplina é objeto de estudo. O resultado positivo pode ser avaliado ao final do curso, quando os alunos, ainda motivados, continuam a colaborar com a página da disciplina, mesmo depois de passados vários anos de conclusão da mesma.

Segundo os autores, observa-se que o estudante, motivado, acaba por adotar uma postura mais ativa, independente e responsável, possibilitando ganhos relacionados a habilidades de comunicação oral e escrita, capacidade de pesquisa e comportamento ético, dentro de seu próprio estilo. Já o professor, por sua vez, abre mão do papel de autoridade para se tornar um conselheiro, guia do grupo, orientador, facilitador, mediador e é também exigido em termos de participação na busca da melhor solução.

Pode-se concluir que a troca dinâmica e com grande quantidade de informações que caracterizam processos deste tipo enriquece conteúdos, aprimora professores e alunos e torna mais motivante a experiência para todos os usuários. O compartilhamento de idéias, a colaboração para melhoria do grupo e da informação disponibilizada, os relatos de experiências individuais, além da mudança de papéis torna EVI uma experiência única para os envolvidos.

Por outro lado, a rápida valorização da Internet e de programas computacionais como aliados ao processo educacional, esconde uma diversidade de problemas e falsas soluções.

Visando a análise de tais problemas e soluções, autores como Bork; Britton (1998), Aragão et al. (2000) e Maçada; Basso (2000) apontam algumas decepções, uma vez que grande parte de cursos de EVI não provê (ou provê muito pouca) interação e identificam que há grande risco (na avaliação de resultados), sobretudo quando a Internet é usada como fator principal de ensino.

Esta crítica reforça a falsa expectativa de que a simples utilização de uma tecnologia é a solução para os problemas, apontando que informatizar o material tradicional (anteriormente aplicado em educação presencial), sem uma adequada alteração das técnicas de ensino, não é solução por si só (Robles et al., 1997). O risco declarado consiste em confundir a entrega de informação com aprendizado, alijando elementos essenciais, tais

como resolução de problemas, criatividade e imaginação dos instrutores e dos alunos (Bork; Britton, 1998).

Assim, segundo Bork;Britton (1998) e Lawhead (1997) são desvantagens do uso inadequado dos meios:

(a) uma fraca interação, sem a qual qualquer esforço de aprendizado pode ser perdido. Enquanto a linguagem natural, presente no ensino face a face envolve séculos de refinamento no sentido de melhoria da comunicação, a interação na rede é limitada;

(b) falta de tratamento particular para cada estudante, sendo que um grande número de pessoas diferentes e com diferentes graus de dificuldades são tratadas de forma similar;

(c) problemas relacionados com a rede em si, envolvendo dificuldade de tráfego; inacessibilidade para alguns e tempo excessivo de carga, além da impossibilidade da comunicação bi-direcional para alguns sítios;

(d) custos de equipamento elevado;

(e) gerenciamento de problemas relacionados a montagens de sítios, tais como violações de direitos autorais;

(f) incerteza do livre acesso a Web ser continuado: será que o livre acesso a Web não tem prazo para terminar?

A discussão pressupõe que a adequação dos meios (Internet, ferramentas de desenvolvimento, projetos de concepção, propostas do que pode ser disponibilizado de forma virtual interativa etc) de forma a propiciar uma melhoria da aprendizagem, quer seja exclusivamente presencial, quer seja virtual interativa, deve ser baseada em análise da necessidade e em projeto direcionado a soluções específicas, sob pena de expectativas frustadas por má soluções (Lawhead, 1997).

Este trabalho investe no desenvolvimento de ferramenta para EVI com a aplicação de técnicas de Realidade Virtual com o objetivo de minimizar os problemas dos itens (a), (b), (d) e (e) citados anteriormente. A seção seguinte destaca aspectos relevantes de tecnologias associadas a Realidade Virtual que justificam desenvolvimentos nesta direção.

2.2. Realidade Virtual

A expressão Realidade Virtual - RV - possui diversas definições, devido à natureza interdisciplinar da área e da sua evolução. Segundo Kirner (1997), pode-se definir

Realidade Virtual como uma forma das pessoas visualizarem, manipularem e interagirem com computadores e dados extremamente complexos, na qual idéias como imersão, interação e envolvimento com o ambiente virtual são consideradas básicas e fundamentais. Uma das principais vantagens desta tecnologia é o envolvimento amplo de sentidos do ser humano na interação homem-máquina.

Desde as primeiras proposições, com a simples utilização de exibição de vídeos até as mais modernas formas de implementação de RV, diversas têm sido as aplicações de tais técnicas, como: entretenimento, treinamento, apresentação de protótipos e educação (Encarnação; Gobel, 1994); (Sherman et al., 1997). As razões desta ampla diversidade de aplicações estão fundamentadas em diversos aspectos. Alguns deles são discutidos nas próximas linhas.

Considerando sons e imagens, usando RV pode prover geração dinâmica, propiciando grande envolvimento do usuário e experiências mais realísticas. Ao mesmo tempo, podem ser usados equipamentos especiais (que podem verificar vários movimentos do corpo do usuário), de forma a prover maior envolvimento e interação.

Com o uso de ambientes virtuais é possível permitir a exploração pelo usuário de forma particular e que atenda suas expectativas com respeito à investigação. Assim, RV pode possibilitar visitas a lugares imaginários, modificando tamanhos, tais como a exploração de um universo imaginário ou da estrutura de um átomo, levando o usuário a forte inserção no contexto de um dado tópico ou assunto, sem riscos, como os presentes no manuseio de equipamentos físicos ou componentes químicos perigosos (Pinho, 1996).

Além disso, sistemas de RV são úteis na possibilidade de manuseio a distância de equipamentos, no controle de força, permitindo que o usuário possa comandar e exercer forças remotamente, sem estar imerso em locais perigosos ou com altos índices de poluição.

A possibilidade de reuniões virtuais e encontro temporal entre pessoas separadas no espaço, por meio de técnicas de ambientes virtuais compartilhados, pode propiciar a redução de custos em aplicações corporativas, encontros inesperados e a formação de comunidades virtuais com forte comunicação síncrona e abstração de ambientes reais.

Visando a aplicação em educação, o alto potencial de desenvolvimento de visualizações pode ser destacado como uma das principais importâncias de RV.

Ferramentas de RV, destinadas a educação, favorecem a fixação de conteúdos a curto e longo prazo e criam uma nova forma individualizada de representação de conhecimentos, propiciando melhor tratamento das informações (Pinho, 2000a). Esta nova forma de pensamento e processamento de informações possibilita um melhor estabelecimento de conexões, relações, soluções de problemas e efetiva memorização.

2.2.1. Justificativas ao uso de Realidade Virtual na Educação

A utilização de RV com fins educativos tem merecido destaque e tem sido avaliada de forma intensiva nos últimos anos (Pantelidis, 1996). Os resultados destas avaliações mostram ganhos, em termos de aprendizagem superiores a diversas outras formas de interação visando educação mediada por computador. Alguns relatos são destacados a seguir.

Byrne (1996) demonstrou que, estudantes do segundo grau, utilizando aplicativos baseados em realidade virtual para análise de experiências de Química (relacionadas com visualização e manuseio de moléculas) apresentaram uma retenção de informações (após três meses) muito superior a estudantes que obtiveram tais informações através de outros meios, tais como sistemas audiovisuais, demonstrando que um dos principais fatores envolvidos com a aprendizagem é a interatividade proporcionada pelo ambiente. Este aspecto é apontado por Costa (2000) apud Rizzo et al. (1997), confirmando que a interação é a característica chave que distingue uma experiência em RV de uma experiência de, por exemplo, assistir a um filme.

Pausch et al. (1997) destaca que usuários de RV são muito melhores nas buscas sistemáticas da informação porque têm lembranças melhores daquilo que olharam na cena que os envolve. Pinho (1996) apresenta o consenso de que a mesma pode influenciar positivamente o processo de aprendizado, sendo que uma das principais justificativas, à esta influência, está na forma de aprendizado, que pode ser baseada em experiências de 1ª pessoa.

Experiências de 1ª pessoa são aquelas na qual o indivíduo conhece o mundo através de sua interação com ele, sendo caracterizado como um conhecimento direto, subjetivo e freqüentemente inconsciente (o aprendiz não tem a clara definição que está aprendendo). Tais experiências são naturais e, geralmente, privadas.

Por outro lado, experiências de 3ª pessoa são aquelas na qual o aprendiz ouve o relato de uma experiência ou aprende a partir da descrição feita por outra pessoa. Esta forma de aprendizado é objetiva, consciente e implícita. Como RV permite a imersão e a exploração individual, o aprendiz vive experiências de 1ª pessoa e explora a informação como uma experiência diária.

O Apêndice II destaca análises, fundamentadas em experiências práticas de aplicações de ambientes virtuais na educação, sobre vantagens de utilização de Realidade Virtual levando em conta as diferentes formas de aprendizado.

Conclusivamente, Bell; Foglerl (1995), Pinho (1996) e Meiguins; Behrens (1999) apontam como principais vantagens da utilização de técnicas de RV para fins educacionais, os seguintes itens:

(a) motivação de estudantes e usuários de forma geral, baseada na experiência de 1ª pessoa vivenciada pelos mesmos;

(b) grande poderio de ilustrar características e processos, em relação a outros meios multimídia;

(c) permite visualizações de detalhes de objetos;

(d) permite visualizações de objetos que estão a grandes distâncias, como um planeta ou um satélite;

(e) permite experimentos virtuais, na falta de recursos, ou para fins de educação virtual interativa;

(f) permite ao aprendiz refazer experimentos de forma atemporal, fora do âmbito de uma aula clássica;

(g) porque requer interação, exige que cada participante se torne ativo dentro de um processo de visualização;

(h) encoraja a criatividade, catalisando a experimentação;

(i) provê igual oportunidade de comunicação para estudantes de culturas diferentes, a partir de representações;

(j) ensina habilidades computacionais e de domínio de periféricos.

Experiências de utilização de sistemas que utilizam técnicas de Realidade Virtual têm sido desenvolvidas e aplicadas nos mais diversos campos de ensino, desde aplicativos

para matemática básica (Keller; Schreiber, 1999), experimentos virtuais de Óptica Geométrica (Kiril et al., 2000) até simulações de circuitos integrados (Ost et al., 1999).

2.2.2. Modalidades de Interação em RV

Ambientes virtuais podem ser concebidos contemplando a utilização de diversos equipamentos, como capacetes, luvas de dados, monitores de vídeo tradicionais, ou ambientes de visualização de diversas vistas, envolvendo o usuário em um panorama imaginário, como os CAVE´s. O apêndice I apresenta equipamentos disponíveis para concepção de ambientes virtuais, com imagens e descrição. A seleção de tais equipamentos acaba por definir as diferentes modalidades de interação.

As modalidades de interação em ambientes virtuais podem ser divididas nos seguintes grupos distintos (Costa, 2000), (Pinho, 2000a):

I) Imersão subjetiva (ou RV não-imersiva): obtida com o uso de monitores e programas que simulam e exploram as projeções perspectivas e transformações geométricas, dando ao usuário a impressão de estar imerso em um ambiente virtual. Perífe ricos comumente usados: monitor de vídeo, teclado, mouse, joystick;

II) Imersão subjetiva com projeção do usuário: o usuário se vê dentro do ambiente virtual que está projetado em um monitor de vídeo através de sua imagem, que é incluída no AV, também qualificada como RV projetada;

III) Imersão espacial: onde o usuário parece estar dentro do ambiente através do uso de dispositivos de imagens e sensores de posição/movimento acoplados ao corpo e capazes de possibilitar a interação direta com o ambiente. Neste caso, eventos ocorrem em todas as direções e consideram os movimentos do usuário. Periféricos comumente usados: capacetes, luvas de dados, sensores de movimento, salas de projeção;

IV) Imersão espacial em CAVE: extensão da anterior, permite a um grupo de usuários participarem simultaneamente de um ambiente virtual, através de projeções de imagens em superfícies que envolvem o(s) usuário(s);

V) Telepresença: o usuário atua, através de instrumentos, em outros locais por meio de operações realizadas no mundo real. Neste caso, a visualização geralmente se dá através de câmeras de vídeo.

Algumas variáveis geralmente utilizadas na definição de periféricos de visualização de ambientes virtuais envolvem resolução, campo visual, qualidade de imagens etc. No entanto, outras características como conforto visual, mobilidade, privacidade e custo merecem ser levadas em consideração no projeto (Bryson et al., 1996).

Fatores psicológicos e de adaptação, além de variáveis de custo, podem levar à escolha do periférico mais adequado à uma cada aplicação. Bryson et al. (1996), Lewis; Griffin (1997) apresentam que Capacetes (HMD), por exemplo, são geralmente rejeitados para trabalhos muito duradouros e técnicos (principalmente aqueles relacionados com a utilização por engenheiros), primeiro porque usuários tendem a relacioná-lo com jogos de computadores e segundo por causarem fadiga visual e até mesmo tonturas quando o tempo de utilização é longo.

Uma das alternativas razoáveis ao desenvolvimento de interfaces inovadoras para RV está relacionada com a utilização de monitores de vídeo (imersão subjetiva). Nesta hipótese, interfaces que provêm ambientes virtuais podem ser desenvolvidas com uso de linguagens como o VRML ou o Quicktime VR.

Johnson (1998) ressalta que uma acurada análise de benefícios providos por tais interfaces precisa ser levada em conta durante o projeto do sistema. Por exemplo, a Internet provê milhares de aplicações desenvolvidas com modelos em VRML que não apresenta muito mais que as capacidades de utilização da linguagem, suportando apenas tarefas triviais (caracterizando forte limitação das tarefas de usuários).

Alguns fatores que justificam o desenvolvimento deste tipo de interface baseiam-se no custo de equipamentos e na facilidade de concepção de aplicativos. Byrne (1996) demonstrou, através de diversas experiências, que o desempenho final de dois grupos de alunos, um deles imerso em um ambiente virtual (imersão espacial) e outro de alunos que usavam a mesma aplicação com imersão subjetiva, foi o mesmo. Além disso, houve preferência pela utilização da tela plana em detrimento ao uso do capacete. Costa (2000) apresenta outras experiências do gênero e aponta que a utilização de monitores não interfere negativamente nos resultados de desempenho dos usuários, gerando resultados promissores, sendo vantagens: utilizar uma tecnologia já sedimentada, evitar as limitações técnicas geradas pelo uso do capacete HMD e outros dispositivos considerados pouco explorados, e aproveitar sua facilidade de uso.

Um aspecto importante na concepção de ambientes virtuais é que a caracterização de usuários potenciais é, geralmente, alijada do processo de discussão. Assim, usuários com perfis diferenciados (como idade, prévio acesso a computadores, estilo de aprendizado, qualificação) acabam por serem considerados de forma única.

Como solução a estes problemas, o desenvolvimento direcionado a domínios de aplicação, com forte caracterização do cliente pode limitar o que se deseja do sistema, ao mesmo tempo que amplia a condição de utilização por usuários do domínio - facilitando a análise de tarefas do sistema e envolvendo o usuário no ciclo de desenvolvimento. Logicamente, os desenvolvedores precisarão levar em conta as diversidades e os problemas relacionados à interação que caracterizam o grupo de potenciais usuários.

Acresce-se a este problema o fato de que o desenvolvimento fundamentado em RV para monitores provê menos interatividade que outras formas de RV. Johnson (1998) propõe que o desenvolvimento de AV fundamentados em uso de monitores de vídeo identifique as tarefas através da observação de operações similares processadas no domínio dos usuários.

Conclui-se que o simples desenvolvimento de ambientes virtuais como ferramentas educacionais, sem prévia análise de domínio e de usuários envolve o risco da concepção de ambientes restritos em tarefas e que possam ser desmotivadores. Assim, é importante a participação dos clientes potenciais nos processos de concepção e validação de protótipos do sistema a ser desenvolvido, visando a coleta de opiniões que propiciem a melhoria das interfaces e das formas de interação.

2.3. Laboratórios Virtuais

Segundo Souza; Oliveira (2001), o conceito de Laboratórios Virtuais engloba duas classes distintas: Laboratórios Remotos e Laboratórios com Conteúdo Exclusivamente Virtual. Componentes das duas classes podem ser acessados remotamente, via rede de computadores.

2.3.1. Laboratórios Remotos

Há um conjunto de instrumentos de simulação de experimentos que está conectado a um computador e este está ligado à Web. Neste caso, o computador controla os equipamentos a ele conectados e provê o acesso remoto a outros computadores.

O experimentador acessa o laboratório remoto (podendo exercer controle sobre o computador e os equipamentos) e faz observações, simulações, testes e coleta de dados de um experimento, que pode estar sendo conduzido por ele ou por outro usuário da Internet. Nesta abordagem, câmaras de vídeo podem ser associadas de modo a exportar imagens para a Web.

Com a utilização desta tecnologia é possível inclusive a conexão de diversos laboratórios simultaneamente, além de permitir que instituições remotas e sem acesso a um dado equipamento (de elevado custo, por exemplo) possam partilhar informações e resultados de experimentos.

2.3.2. Laboratórios com Conteúdo Exclusivamente Virtual

Um laboratório com conteúdo exclusivamente virtual não dispõe dos equipamentos associados ao computador, citados no item anterior. De fato, o(s) computador(es) apresenta(m) programas computacionais capazes de simularem um dado experimento e permitir a coleta de resultados por usuários da Internet.

Aqui, a principal característica é a ausência de um laboratório real, onde se realizam experiências. As simulações, modelagens e visualizações são de conteúdo virtual. Destacam-se como vantagens dos laboratórios com conteúdo exclusivamente virtual:

i) redução da necessidade de equipamentos para as simulações;

ii) possibilidade de, complementar o aprendizado e ser utilizado para experimentar da mesma forma que ocorreria em um laboratório real;

iii) capacidade de simular experiências de grande complexidade;

iv) capacidade de permitir que um experimentador desmonte, visualize e associe peças de difícil (ou impossível) manuseio prático, seja por questões de custo, risco ou de ordem prática de elaboração;

v) possibilidade de uso em atividades onde o tempo ou o custo de produção de um experimento tornariam a montagem real da mesma inviável.

Utilizando Realidade Virtual, a concepção de tais laboratórios pode gerar ambientes virtuais atrativos e com grande similaridade a ambientes relativos àqueles existentes em laboratórios reais. Neste caso, tais laboratórios podem se utilizados na forma de preparação para uma aula presencial, como complemento ao conteúdo ministrado pelo professor ou simplesmente como um recurso que permita ao aprendiz interagir com um dado experimento e colher informações úteis ao seu aprendizado.

O capítulo seguinte explora alguns exemplos de sistemas utilizados no ensino/aprendizagem relativos aos tópicos apresentados neste capítulo.

3. Ferramentas e Sistemas para Educação Virtual Interativa

Este capítulo apresenta algumas ferramentas usadas para desenvolver e suportar conteúdos de EVI e trabalhos relacionados com a proposta desta tese, destacando as principais características e limitações de cada um deles. Algumas conclusões são obtidas a partir da análise dos mesmos que sustentam os principais aspectos da proposição de novas arquiteturas.

3.1. Linguagens para Suportar Conteúdos de EVI através da Internet

3.1.1. Java

A linguagem Java foi projetada para ser o padrão universal na transferência de conteúdos dinâmicos e executáveis através da Web (Ritchey, 1997). Do ponto de vista do programador, a facilidade provida pelo desenvolvimento voltado a um único equipamento: a *Java Virtual Machine (JVM)*, permite que pequenos grupos de programadores (ou um só) possam concorrer com suas aplicações.

Do ponto de vista de provedores, Java pode ser implementada nos servidores normais de HTTP, sendo servida em páginas da WEB como se fossem arquivos de imagens.

Java é usada para desenvolvimento de programas independentes, na forma de aplicativos ou para criação de *applets*, que, associados a páginas da WEB executam uma dada tarefa no ambiente hospedeiro, o navegador.

As *applets* no entanto, apresentam algumas limitações, que variam de acordo com a velocidade da rede, processador local, navegador:

(a) por serem programas, os navegadores necessitam impor uma série de restrições de segurança que dificultam o desenvolvimento de aplicações interativas;

(b) não podem ler ou escrever arquivos na máquina cliente, criando dificuldades ao desenvolvimento de aplicações interativas;

(c) não têm acesso a algumas informações do sistema, tais como memória disponível, podendo causar infrações ou violações;

(d) *applets* podem ser extensos e demandarem muito tempo de transmissão e carregamento.

O ciclo de desenvolvimento de um *applet* envolve a criação do código fonte e dos arquivos HTML para ativá-los, a compilação dos fontes em bytecodes, o teste dos bytecodes com um *appletviewer* ou navegador, a detecção e remoção de erros (com uso de um depurador) e a transmissão da *applet* para o diretório correto no servidor de WEB (Chan et al., 1999).

Para manuseio de imagens, a API Java2D estende os mecanismos do pacote java.awt, permitindo a manipulação e visualização de imagens bi-dimensionais, além de textos, cores, imagens e fontes. A API Java2D permite desenhar formas padrões, como retângulos, círculos, vetores etc e formas arbitrárias, como curvas de Bezier, realizar transformações gráficas sobre formas e preenchê-las (até mesmo com gradientes).

Java também permite manusear formas 3D através de Java3D (que não é uma API), que tem recursos de criação e manipulação de formas tri-dimensionais, inclusive permitindo a renderização de formas, que necessita ser instalado separadamente e tem um código nativo (não independente de plataforma), o que pode ser um fator dificultador de desenvolvimento de interfaces e de utilização na Internet.

Outra relevante característica de Java, associada à Internet, é a capacidade de manuseio dos elementos do protocolo básico da rede (TCP/IP). Classes como *InetAddress* (usada para armazenar endereços da rede), que fazem parte do pacote *java.net* podem ser utilizadas para controlar a exibição de páginas a partir dos *applets*.

Além disto, *applets* podem ser usados para lerem arquivos na Internet (JavaBrasil, 2000), gerando arquivos locais com tais informações, criar conexões entre arquivos através de soquetes, (Java, 2000), acessar serviços de servidores *Proxy*, enviar mensagens eletrônicas usando um servidor *smtp (send mail transfer protocol)*, implementar sítios de bate-papo etc (Ritchey, 1997); (Chan et al., 1999).

Razões como possibilidade de manuseio de imagens bidimensionais de forma simples, disponibilizando pela Internet conteúdos que podem ser úteis ao aprendizado tem

levado diversos programadores a conceberem *applets* que se prestam a ilustrar conceitos e demonstrações, permitindo seu uso como ferramenta para a educação virtual interativa.

Labvirt (2000) é um exemplo desta forma de aplicação que pode ser visualizado no sítio do Instituto de Física da Universidade de São Paulo (USP). O conteúdo do sítio apresenta uma enorme coleção de *applets* Java capazes de apresentar simulações de conceitos e leis da Física.

Uma possibilidade de aplicações relacionadas a Java e Internet para fins educativos, refere-se ao desenvolvimento de ambientes cooperativos - CSCW - *Computer Supported Cooperative Work*, que permitam a concepção de grupos de trabalho com auxílio de computadores.

Bom exemplo desta forma de desenvolvimento é o projeto Jets (1998) - *Java Enabled Telecollaboration System*, um sistema totalmente implementado em Java pela Universidade de Ottawa, que pode ser considerado como uma API para o desenvolvimento de *applets* multimídia compartilhados que provê mecanismos de controle de acesso, consistência e passagem de dados e informações. O sistema provê *applets* de *whiteboard* multimídia compartilhados, visualizador VRML e *applet* gerenciador de sessão, de forma que todos os participantes têm acesso aos *applets* compartilhados.

3.1.2. VRML

VRML ou *Virtual Reality Modeling Language* tem sido aplicada em diversos projetos para concepção de mundos virtuais para EVI (Costa, 2000); (Ipolito; Kirner, 2000); (Kiril et al., 2000). A linguagem é uma importante aliada no desenvolvimento de mundos tridimensionais interativos na Web e sua história é fundamentada na colaboração de diversos pesquisadores e importantes empresas relacionadas com a Computação Gráfica (Nadeau, 1999).

VRML é adequado às mais diferentes aplicações, incluindo desde modelagens de superfícies simples como um cone até mapeamentos geográficos complexos de terrenos. A pesquisa envolvendo VRML tem apresentado resultados positivos em aplicações relacionadas às mais diferentes áreas do conhecimento (Cardoso et al., 2000b).

Com a utilização exclusiva de VRML temos exemplos envolvendo aplicações na modelagem de ambientes similares a ambientes reais, como a experiência de

desenvolvimento do campus virtual da UFSCar - UFSCar Virtual, em projeto que comprova a aplicabilidade de VRML para modelagem, com inclusão de texturas, de ambientes reais (Ipolito; Kirner, 1997).

Matsuba; Roehl (1999) constataram a possibilidade de utilização de VRML para apresentações em tempo real, usando-se as formas mais comuns de configurações da Internet hoje, dispensando adequações de rede e arquivos com a concepção e apresentação da peça teatral: "*Shakespeares´*s - *A Midsummer Night´s Dream*", totalmente desenvolvida em VRML, usando atores para simular as vozes e representações de teatro, elementos de cena e personagens. O sítio da apresentação ("http://www.vrmldream.com/") permitiu que diversos espectadores pudessem assistir pela rede mundial uma peça teatral completa.

Chen et al. (1997) relatam a construção pela *Silicon Graphics* do O2, um novo tipo de estação de trabalho multimídia e a incorporação de uma nova experiência, em termos de interface, para introduzir a cada novo usuário a capacidade total de sua nova máquina. Tal interface (multifuncional e multimídia), com documentação, demonstrações e interface específica foi criada com HTML, VRML e Java Script. Esta forma foi uma oportunidade de demonstrar que 3D, usando VRML, é uma técnica de interface para simulações em larga escala e de forma comercial

Lawton (1999) apresenta, dentre outros trabalhos, a concepção que a CNN (2000), uma das grandes corporações de comunicação nos E.U.A., apresenta em seu sítio na Internet de fenômenos meteorológicos usando VRML, imagens de alta resolução e *frames* em HTML. Tais simulações evidenciam a possibilidade de utilização da linguagem para apresentar situações reais e permitir a usuários visualizar fenômenos físicos reais, como um ciclone - Fig. 3.1.

Reddy et al. (1999) exploraram o potencial de VRML para mapeamento de terrenos através do projeto "Terra Vision II", composto por mapas 3D com dados espaciais sobre terrenos. Os autores apontam como grandes vantagens da linguagem e que propiciaram o desenvolvimento do projeto: escalabilidade, flexibilidade de representações 3D e suporte a malhas tridimensionais.
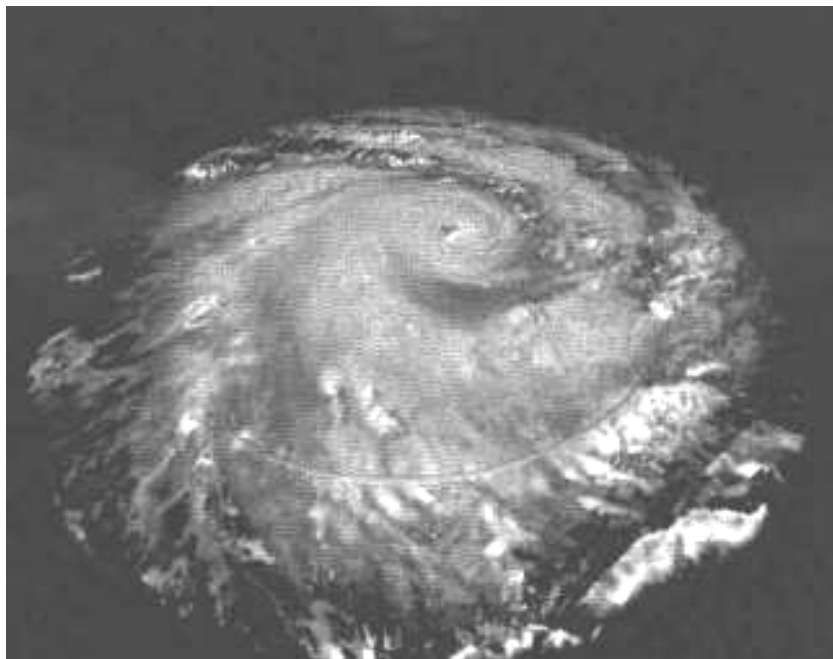
Fig. 3.1 - Visualização de um Ciclone com uso de VRML

Silveira; Pellegrino (1997) desenvolveram aplicações de VRML para modelagem de sólidos aplicadas à construção de mundos virtuais foram desenvolvidas usando a primeira versão da linguagem - VRML 1.0 e evidenciando, desde então, aplicabilidade da linguagem ainda na versão inicial, usando estruturas de dados como *octrees*.

3.2. Laboratórios Virtuais baseados em Java - 2D

3.2.1. Virtual Laboratory - Johns Hopkins University

O Virtual Laboratory (2002) corresponde a um conjunto de *applets* Java usados para visualizar experimentos relacionados com eletrônica, robótica, propagação de ondas, condução do calor e estatística.

Os experimentos são apresentados na forma de simulações bidimensionais que permitem ao usuário: inserção de elementos, alteração de parâmetros dos elementos utilizados e visualização de grandezas resultantes de simulações experimentais.

A Figura 3.2 refere-se a um destes *applets*, usados para concepção de circuitos digitais (circuitos lógicos), onde o aprendiz pode inserir portas "and", "or", "not", "nand" ou "nor" e a partir de quatro entradas (*input*) e visualizar as saídas (*output*) a partir das

possíveis entradas. A combinação das portas é efetuada por ligações físicas estabelecidas com linhas que o próprio usuário define, como pode ser visto na Figura 3.3.
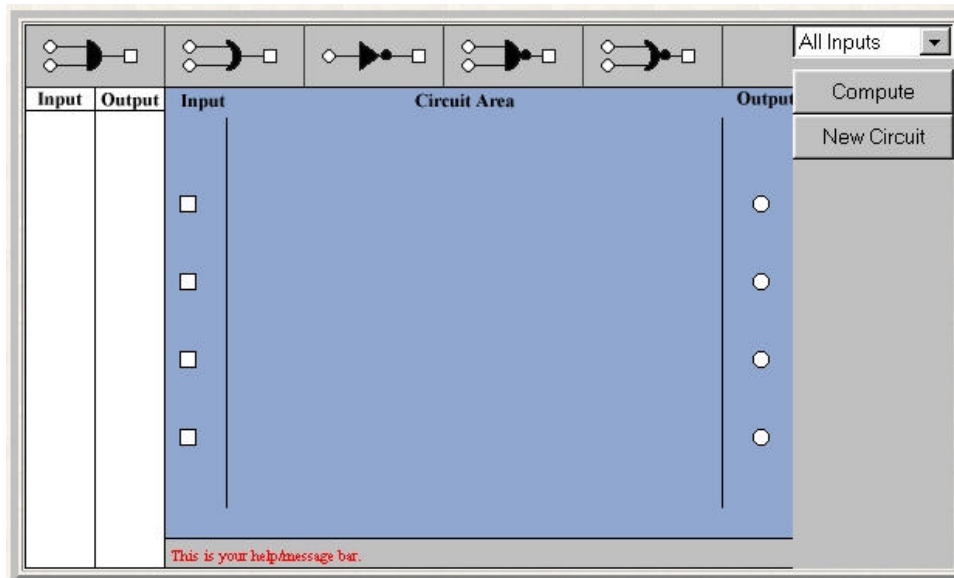
Fig. 3.2 - Applet de Simulação de Circuitos Lógicos
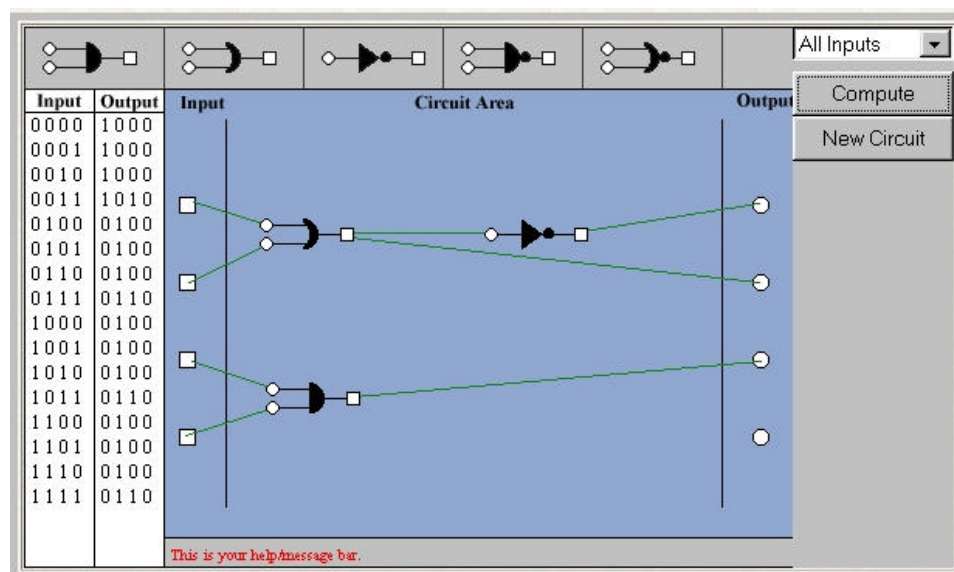
Fig. 3.3 - Resultado do Circuito Lógico Simulado - Entradas x Saídas

Os resultados das simulações são facilmente visualizados pelos usuários. Destacam-se como propriedades relevantes deste laboratório virtual:

i) interfaces amigáveis e interativas: um experimentador insere e define elementos e relações de acordo com suas necessidades;

ii) fácil visualização de resultados, que são apresentados de forma rápida nas diversas simulações suportadas pelos *applets*;

iii) portabilidade: acessível via Web, os experimentos podem ser utilizados com grande facilidade;

Destacam-se como principais limitações:

i) número reduzido de experimentos;

ii) impossibilidade de um professor prover inserção de novos elementos, distintos dos disponibilizados no *applet* e impossibilidade de fazer alterações nos existentes;

iii) erros na definição de elementos tornam o experimento inválido e forçam o usuário a reiniciar todo o processo de definição.

3.2.2. *Physics Applets* - Universidade de Oregon

Disponibilizado pela Universidade de Oregon (VLAB - Oregon, 2002) e desenvolvido pelo departamento de Física, este laboratório virtual é constituído de um conjunto de *applets* destinados exclusivamente a tópicos da Física.

Há um grande número de experimentos, das mais diferentes frentes da Física (Astronomia, Energia, Mecânica e Termodinâmica). O usuário escolhe, dentre uma lista de assuntos (Figura 3.4) um dado *applet* e pode interagir com objetos do mesmo para definir uma dada simulação de um tópico particular da Física.

O exemplo apresentado pelas Figuras 3.5 e 3.6 refere-se à simulação de movimento de um corpo sob ação de uma dada aceleração. O *applet* traz informações gráficas (o gráfico é traçado dinamicamente) sobre variáveis da Física relevantes a tal tópico (velocidade e posição), além de permitir que o usuário defina os valores da posição inicial, velocidade inicial e aceleração.
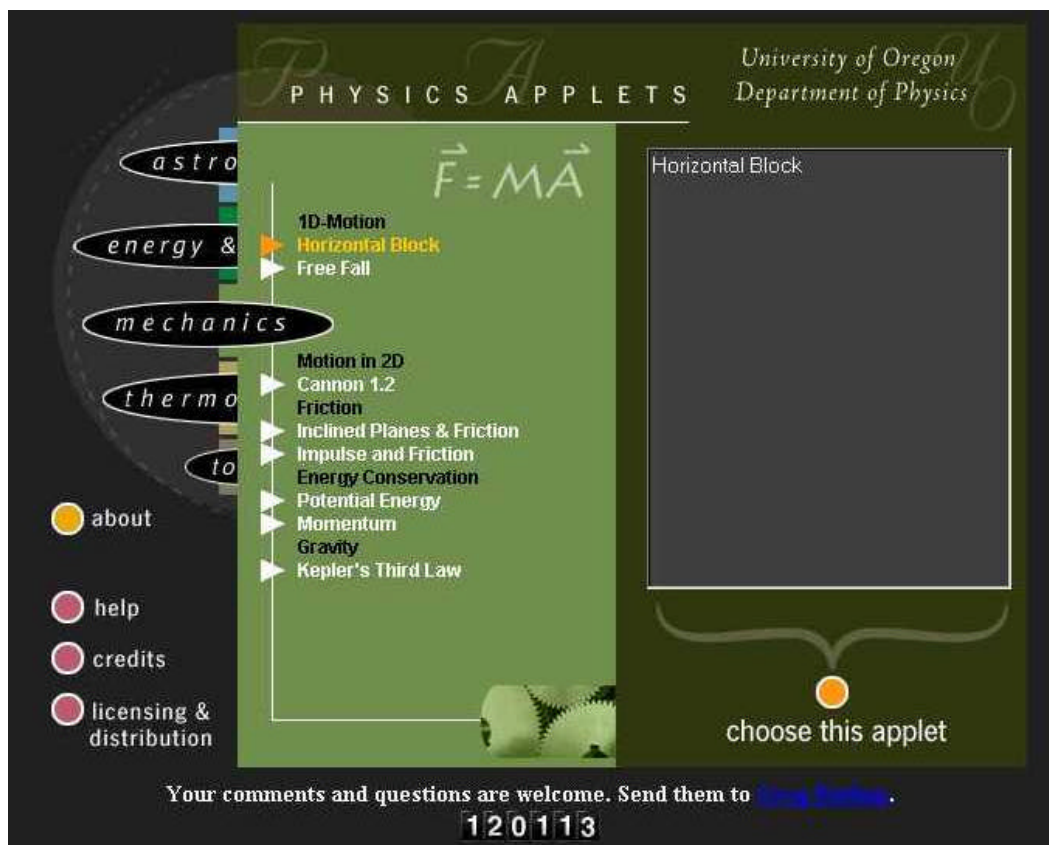
Fig. 3.4 - Página de Escolha do Experimento - Physics Applets

São características comuns dos experimentos virtuais:

- simulações bidimensionais;
- *widgets* que permitem ao usuário variar parâmetros, iniciar e coletar informações de resultado de um dado experimento;
- impossibilidade de alteração do cenário que constitui o experimento, seja por inserção ou remoção dos elementos.

A Figura 3.5 apresenta o aspecto inicial do *applet* disponível relativo à movimentos uniformemente acelerados. Verifica-se que o usuário, neste experimento, pode alterar os valores de posição inicial (P), velocidade inicial (V) e aceleração (A), executar o experimento (Run) ou reiniciar o processo (Reset). O resultado de uma simulação pode ser visualizado na Figura 3.6, onde o móvel tem sua posição alterada e os gráficos retratam a posição (gráfico à esquerda) e a velocidade (gráfico à direita) em função do tempo.
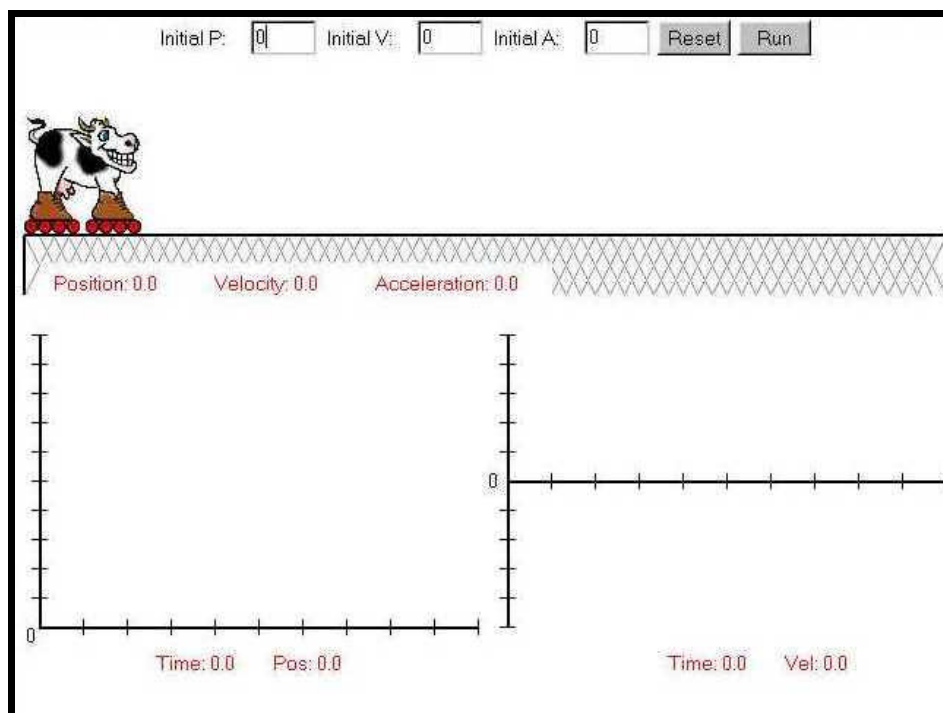
Fig. 3.5 - Simulação de Cinemática - Physics Applets
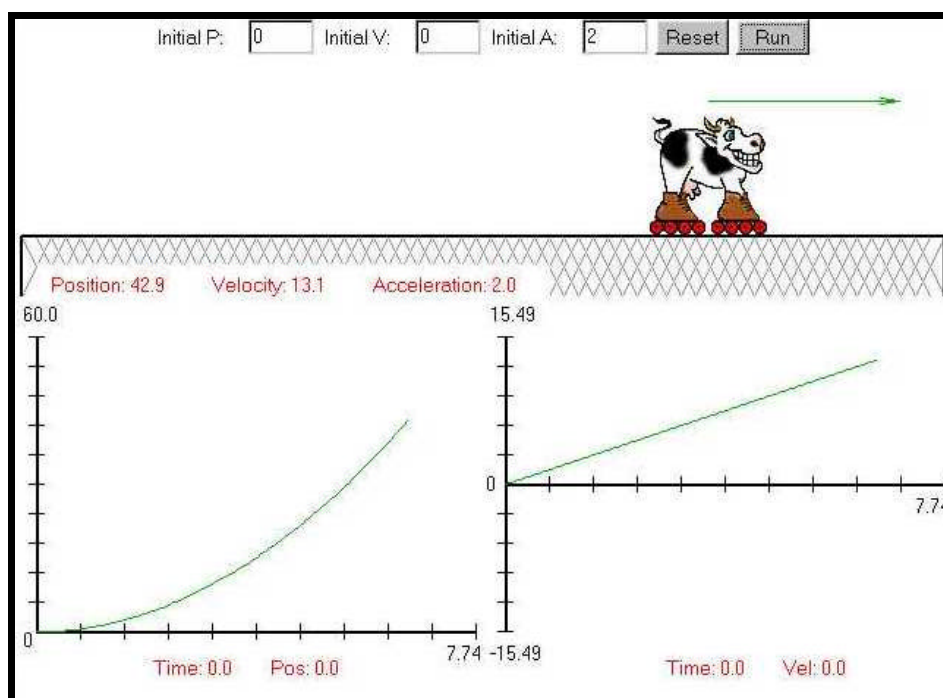
Fig. 3.6 - Resultado da Simulação

Como pode ser observado na Figura 3.6, um vetor posicionado sobre o móvel indica a existência de velocidade final. Outras informações são dadas pelo aspecto e valores plotados nos dois gráficos exibidos. Da mesma forma que o *applet* apresentado neste exemplo, há outras simulações disponíveis que seguem os mesmos princípios de construção e operação.

3.2.3. LabVirt - Laboratório Virtual de Física da USP

O laboratório virtual de Física da USP (Labvirt, 2000) refere-se a um projeto de uma fábrica de animações para a Web associada à uma série de links e animações existentes - *applets* e conta ainda com recursos de comunicação assíncrona com professores e interessados na área.

O sítio contém um grande conjunto de animações em Java e links para diversos sítios no Brasil e Exterior que apresentam outros *applets* ou animações em VRML e *Shockwave*, provendo condições de o usuário solicitar uma consulta a um físico da USP - por *email*, além de seções relativas a fóruns de discussão, indicação de projetos educacionais e artigos selecionados.

Com o acesso às animações, são visualizados *applets* Java relativos a conteúdos específicos de Física . As Figuras 3.7, 3.8 e 3.9 apresentam uma simulação, através de um *applet* disponibilizado pela página, da Segunda Lei de Newton, contemplando a união dos conceitos de massa, força e aceleração.

Como pode ser visto na Figura 3.7, o *applet* apresenta um conjunto de Forças (*Forces*) e um conjunto de Massas (*Masses*), além de campos relacionados com valores numéricos de Energia Cinética, Aceleração, Velocidade, Distância e Tempo. O usuário deve selecionar a força e a massa que será acelerada e o *applet* se encarrega de apresentar uma animação de movimento da massa, além de dados relativos ao movimento adquirido pela massa acelerada.
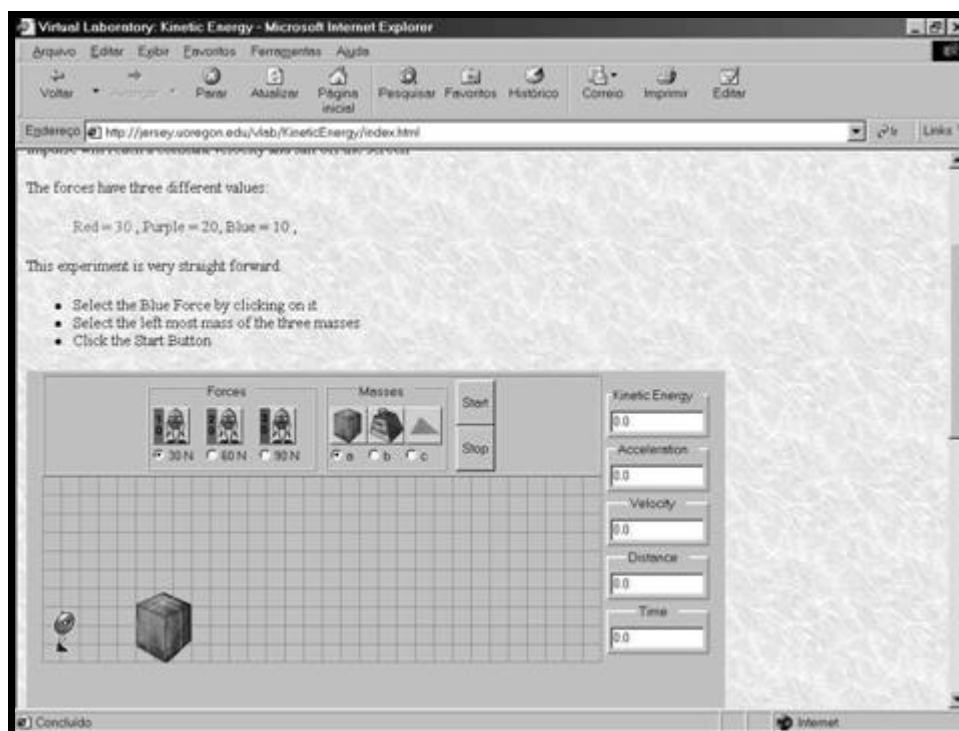
Fig. 3.7 - *Applet* de Simulação da Segunda Lei de Newton - Inicial

A Figura 3.8 apresenta o resultado da simulações usando uma Força de 30N e a massa "a", que é a menor do conjunto de massas. A Figura 3.9 apresenta o que ocorreria com a seleção da Força de 60N e a mesma massa "a". Pode-se observar que o elemento de maior massa realmente ganha menor aceleração, varrendo menos espaço.

As principais limitações deste sistema são:

i) visualizações bidimensionais, não manuseando modelos tridimensionais que se assemelham a objetos físicos reais, ou seja, na simulação tem-se uma foto de objeto, mas, não o manuseio de um objeto que se assemelha a um objeto de laboratório. Esta limitação é comum a representações bidimensionais;

ii) impossibilidade de autores criarem seus próprios experimentos, modificando a versão disponível de acordo com a necessidade.
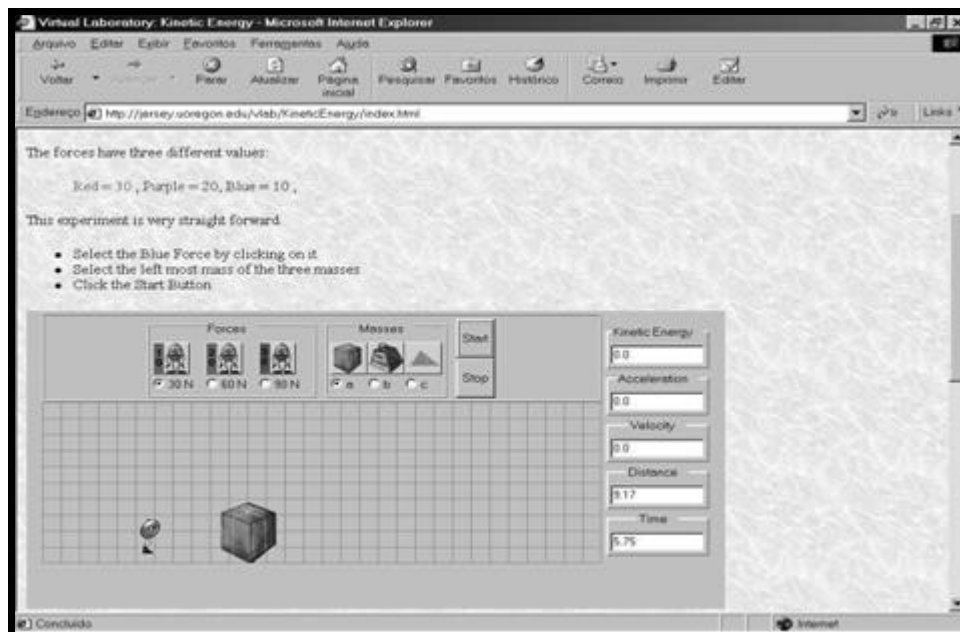
Fig. 3.8. Resultado da Aplicação de Força de 30N na Massa "a"
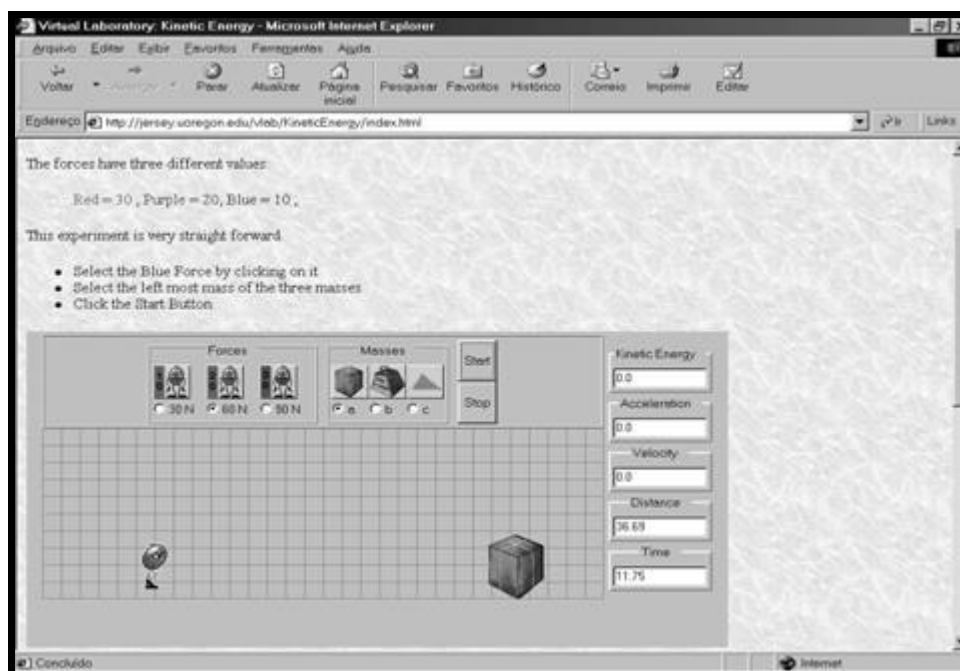
Fig. 3.9 - Alterando a Força: Nova Simulação e Novos Resultados do *Applet*

3.3. Laboratórios Virtuais e Sistemas de EVI em 3D

Diversas soluções são apresentadas neste tópico, relativas ao desenvolvimento de sistemas e laboratórios virtuais caracterizados por interação em ambiente 3D. Sistemas como Alice e o projeto *ScienceSpac*, caracterizados pelo envolvimento de um grande número de pesquisadores são relatados.

Também são apresentadas soluções fundamentadas na utilização das linguagens HTML, Java, VRML e JavaScript, caracterizadas pelo desenvolvimentos de páginas WEB interativas, capazes de permitirem ao usuário experimentar e visualizar conceitos educacionais. Soluções de combinação destas linguagens têm permitido a visualização de experimentos físicos e químicos, elementos geométricos, tópicos de geografia de terrenos, observações espaciais e ambientes de discussão síncrona para fins de aprendizado a distância.

3.3.1. Alice

3.3.1.1. Características

Alice (2000) é uma ferramenta para criação e animação de mundos tridimensionais, definida para ser fácil de programar e modificar. Os autores visaram o desenvolvimento, que iniciou-se na *"University of Virginia"* e posteriormente continuou com uma equipe na *"Carnegie Mellon University"*, de um programa que pode ser utilizado por pessoas que não sabem programação, evitando as etapas demoradas de aprendizado para utilização de tecnologias desta natureza (Pierce et al., 1997).

Alice foi desenvolvido, desde o seu projeto inicial, em *Python*, uma linguagem interpretada escrita por *Guido Van Rossum*. *Python* é uma linguagem de alto nível com uma sintaxe simples e de fácil aprendizado, orientada a objetos, sendo comparada por alguns observadores a um pseudocódigo executável (Python, 2000).

Por ser interpretada, *Python* propicia facilidades na alteração e validação de código, sendo que mudanças efetuadas podem ser observadas em segundos, e, por esta razão é mais rápida que C ou C++ para prover alterações, ao mesmo tempo em que é extensível a C e C++.

De forma a evitar que usuários necessitassem estudar e aprender conceitos relacionados a computação gráfica ou realidade virtual, como transformações (escala,

rotação, translação, matrizes, coordenadas homogêneas etc), Alice provê suporte a requisitos de computação gráfica e realidade virtual, sem a exigência do aprendizado destes conceitos.

Ademais, usuários podem ser levados a cometer erros de sobreposição de transformações, como uma aplicação de transformação de escala, que pode levar à modificação da figura visualizada, de uma forma não intuitiva. Alice pretende evitar que o desconhecimento destes conceitos seja fator desmotivador.

O programa possui um ambiente de criação e desenvolvimento, compatível com Windows 95/98® e Windows NT 4.0® (requer instalação do Service Pack 3.0) e um *plug-in* para ser associado aos principais navegadores da WEB, de forma a possibilitar a visualização de mundos virtuais desenvolvidos no ambiente de criação por usuários da rede mundial de computadores.

Para ser executado, Alice requer máquinas Pentium® ou superiores, não necessitando de placas aceleradoras gráficas, mínimo de 16M de memória, placa de som, placa de vídeo VGA Graphics e instalação do *Microsoft Direct 3D*®(biblioteca gráfica do *Microsof Direct X*®).

Há diversos objetos 3D previamente definidos, caracterizados por atributos e propriedades, alguns herdados da classe *object* do Alice, que permitem alterações e criações de cenários e dinamismo de cenários.

Comandos de alto nível permitem modificações de atributos, aplicação de transformações gráficas, desenvolvimentos de cenários, movimentação de câmera (alterando o ponto de vista do usuário) e navegação. As *widgets* são de fácil assimilação e suas funções estão relacionadas com os tópicos relatados.

O usuário é levado à navegação através de cliques, apontamentos e arrastes, com a possibilidade de inserir novos elementos, comandar ações de objetos e suas partes, além de poder carregar uma série de exemplos existentes.

A Figura 3.10 apresenta a interface gráfica do programa Alice, versão mais atual, onde podem ser observadas: a janela de visualização de imagens bem no centro da GUI; os elementos de interface de controle da visualização - "*Camera Controls*" - na forma de setas, de modo a facilitar a operação de visualização e exploração; os botões de adição de elementos e textos no cenário - "*Add Object*" e "*Add Text*"; uma linha para comandar ações

sobre objetos seguida de *"Go";* uma árvore que apresenta as atuais definições da cena; o menu de comando.

Fig. 3.10 - Tela Inicial do software Alice

Além disso, uma segunda pasta, denominada *Script* pode ser ativada, de forma a definir um roteiro de animação de objetos (e/ou suas partes). O *"Help"* contém uma seção de exemplos que podem ser ativados para análise, com diversos exemplos de animação, adição de sons, movimentações de partes do objeto etc.

A observação da GUI permite concluir a disposição dos programadores em desenvolver uma interface intuitiva, de forma a possibilitar tanto o desenvolvimento quanto a navegação simplificada.

3.3.1.2. Utilização da Ferramenta

A elaboração de mundos virtuais necessita da definição dos elementos que comporão a cena, a relação entre eles, o fundo, o ponto de vista - câmera, dimensões dos elementos, algumas propriedades físicas, cores etc. Este tópico apresenta a solução proposta por Alice, de forma a permitir o desenvolvimento do propósito do usuário.

A cena é composta por três elementos: *Camera, Light, Ground. Camera* é usado para definição de ponto de vista do usuário, além de posição do mesmo. *Light* é usada para definir a iluminação da cena com e *Ground* para especificar os elementos que estarão na cena - objetos e seus atributos. Cada um dos elementos apresenta uma árvore específica que permite analisar o que está sendo usado.

A GUI apresenta ainda uma janela para a cena, uma linha para associar ações aos elementos - *do,* e diversas *widgets* para exploração, que possibilitam ao usuário andar para frente, para trás, lados, mover para cima e para baixo, olhar acima e abaixo.

Com a utilização de comandos como "*Add Object*" e linhas de comando como "*object.move(up,0.5)*" inseridas na GUI, facilmente se obtém uma cena em que um objeto inserido se move para cima de 0,5m.

A elaboração de cenas dinâmicas com diversos objetos e movimentos associados pode ser feita com o auxílio de *Scripts. Scripts* apresentam a possibilidade de, usando uma série de linhas de comando e uma meta-linguagem coordenar objetos e suas partes e elaborar mundos dinâmicos (onde elementos perfazem uma série de ações previamente definidas).

Para escrever um *Script,* o usuário necessita conhecer os comandos disponíveis e ter uma noção clara de que objetos são formados por partes que podem ser coordenadas isoladamente através de linhas específicas.

Como exemplo, o objeto helicóptero (previamente disponibilizado) possibilita definir: hélice superior - *Top Shaft*, base esquerda - *Left Skid,* hélice posterior -*Back Shaft* e base direita- *Right Skid.* Suponhamos a intenção de simular um helicóptero que aciona as hélices, alça vôo e começa a se mover para frente com uma dada velocidade. A seqüência de instruções necessárias, de forma a movimentar diversas partes do objeto, seria composta por comandos como "*Helicopter.TopShaft.Rotor.pan(left,speed=1.5*)" - para simular a movimentação da hélice superior ou "*Helicopter.BackShaft.Rotor.pan(up,speed = 2.5)*" - para simular a movimentação da hélice traseira.

A união de todas estas linhas é um *Script*, bastando que o usuário as coloque nesta ordem na pasta *Script* e determine sua execução por um clique no botão "*PerformScript*". Objetos podem ainda ser destruídos através do comando "*destroy*" e assim serem removidos da cena.

Para verificar a utilização do programa em concepção de uma aula de Física, desenvolveu-se um estudo de caso de utilização do programa visando apresentar o conceito de aceleração e sua relação com a massa e com a força aplicada - Princípio Fundamental da Dinâmica. Dois usuários distintos serão simulados, o professor que compõe a aula e o aluno (aprendiz).

De forma a analisar a atuação da aplicação de uma força sobre uma massa deve ser observado que sua aplicação resulta em uma aceleração proporcional à força aplicada e inversamente proporcional à massa do corpo que está sendo acelerado.

Assim, o professor necessita de dois elementos: aquele que atuará como aplicador (ressalva: pela Terceira Lei de Newton, qualquer ação corresponde a uma reação) da força e aquele que vai ter sua inércia modificada pela aplicação. Opta então por associar a um objeto "soldado" a atuação de aplicador e a uma pequena scooter a ação de receptor, objeto que será acelerado.

Para compor a cena, adiciona os objetos usando por duas vezes a função "*Addobject*", tomando cuidado para que os dois objetos estejam afastados, mas, alinhados, de forma a permitir a troca de forças entre eles.

Após a inserção dos objeto deve-se compor o *Script* para que o objeto "soldado" seja animado em direção ao objetos "*scooter*", de forma que, com a colisão, o objeto "*scooter*" comece seu movimento, propiciado pela aplicação da força. Uma das possibilidades de *Script* seria a apresentada na Figura 3.11.

linha	Comando
1	`ToySoldier.leftarm.turn(up,0.2)`
2	`ToySoldier.rightarm.turn(up,0.2)`
3	`SoldadoMove = ToySoldier.Move( Forward, duration =3.0 ).Stop()`
4	`ScooterMove = Scooter.Move( Forward, 1.0, speed =0.5)`
5	`Scooter.RespondToCollisionWith( ToySoldier, ScooterMove )`
6 7	`SoldadoMove()` `Camera.PointAt( ToySoldier, EachFrame )`

8	`def ToySoldierReport( obj1, obj2 ):`
	`print obj1, "collided with", obj2`
	`ToySoldier.RespondToCollisionWith(Scooter, ToySoldierReport )`

Fig. 3.11 - Script para Demonstração da Segunda Lei de Newton

As linhas um e dois definem movimentos para os braços do "soldado", de maneira à abstrair situação de levantamento dos braços para aplicação de força. A linha três ordena o movimento do soldado para frente, com duração de 3 segundos.

Na linha quatro, define-se uma função de forma a nortear o movimento do "*scooter*" para frente, com duração de 1.0 s e com velocidade constante de 0.5m/s. Esta função necessita ser ativada para execução e foi denominada "*ScooterMove*".

Para ativar a resposta à colisão, que deve fundamentar a aplicação da força, a linha cinco define que ao acontecer a colisão do "soldado" com o "*scooter*", é ativada a função "*ScooterMove*", que ativa o movimento do "*scooter*".

A linha seis manda ser executado o movimento do soldado, que dispara o processo. A linha sete faz com que a câmera se movimente, acompanhando o movimento do "soldado" a cada quadro(*frame*) da animação. Os comandos da linha oito apenas imprimem que houve uma colisão entre os elementos do cenário - para controle da operação - e não influem na animação em si.

As figuras seguintes mostram a seqüência de movimentos decorrente do *Script* elaborado e os efeitos visuais. Na Figura 3.12 é apresentada a situação inicial.
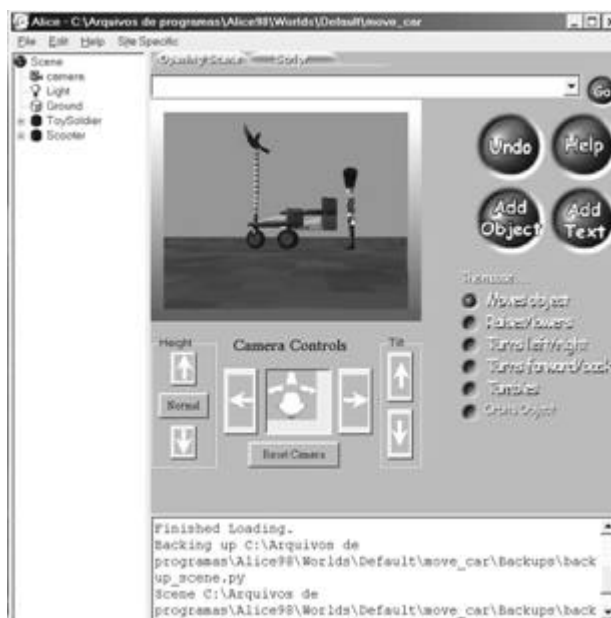
Fig. 3.12 - Definindo Elementos para Simulaç ão de Movimentos

A Figura 3.13 mostra-se o momento do impacto (colisão) entre os elementos e, finalmente, a Figura 3.14 mostra a situação final dos elementos da cena, após o movimento do "soldado" e do "*scooter*".

A simulação corresponde à execução do *Script* apresentado na Figura 3.11 através do acionamento do comando "*PerformScript*" que se encontra na pasta "*Script*" da GUI.
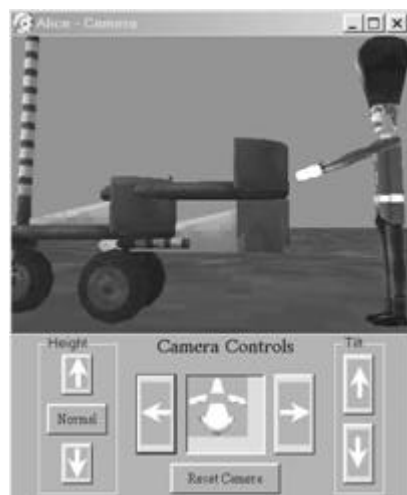
Fig. 3.13 - Instante da Aplicação da força no Objeto "*Scooter*"
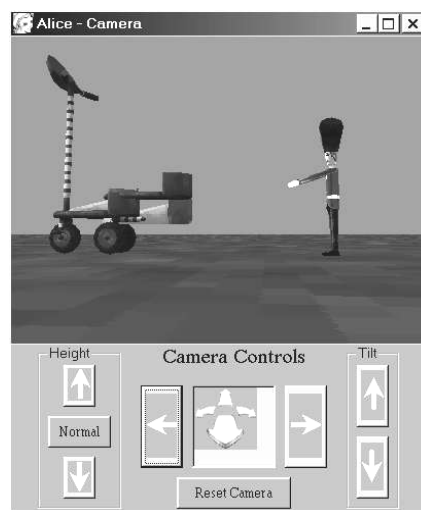
Fig. 3.14 - Resultado da movimentação do "*Scooter*"

A experiência permite constatar:

(a) A necessidade de aprendizado de uma linguagem específica concebida para definir os movimentos dos objetos na cena, que é essencial para a execução do *script*;

(b) A dificuldade em definir comportamentos físicos específicos usando a ferramenta, uma vez que variáveis como aceleração, força, massa não se encontram disponíveis;

(c) Durante o desenvolvimento, o usuário precisa tomar muito cuidado com posição de elementos, uma vez que as definições das mesmas são dadas pela posições da câmara e podem gerar confusão ou uma má definição, no caso do exemplo, se objeto 1 - soldado - não estiver alinhado com o objeto dois - scooter - não haverá a colisão e não será iniciado qualquer movimento do objeto dois;

(d) A facilidade de criar e destruir elementos inseridos no cenário, que com um simples clique podem ser adicionados ou removidos, bem como a facilidade de navegação usando os movimentos do mouse.

Conclui-se que Alice é uma ferramenta útil e de fácil manuseio, com uma interface amigável e que permite a utilização de diversos objetos de forma a desenvolver cenários interessantes e experimentos úteis, que tem como principais limitações:

(a) a dependência de programas computacionais compatíveis com sistema operacional específico - família *Windows*;

(b) a necessidade de aprendizado, pelo usuário, de uma linguagem composta por comandos que fazem uso de conceitos não familiares a usuários de que não estejam acostumados a linguagens de computação gráfica;

(c) o programa não permite que novos objetos sejam inseridos pelos usuários, que fica limitado a utilizar os objetos disponíveis;

(d) como a ferramenta é genérica, sua adaptação a um dado conteúdo cria dificuldades ao usuário final de aplicação específica, como por exemplo, uma que pretenda usar a mesma para demonstrar conceitos de Física.

3.3.2. ScienceSpace*:* Usando RV para Conceitos Científicos

De forma a investigar a aplicabilidade de experiências de laboratório simuladas através de experiências geradas por computadores, o projeto ScienceSpace (Salzman et al., 1995, PSS Science Space Project, 1999) tem sido elaborado desde os meados do ano de 1994, através do desenvolvimento de uma coleção de mundos virtuais que visa auxiliar

estudantes e professores na compreensão de conceitos científicos, principalmente relativos à Química e Física.

Esta coleção de mundos virtuais utiliza hardware relativo a estações de trabalho *Silicon Graphics Onyx* e provê imersão e interação dos usuários através de periféricos como capacetes, equipamentos de som estéreo (*headphones*) e um sistema de rastreamento das mãos e da cabeça do observador. Do ponto de vista de modelagem geométrica, a concepção dos objetos dos mundos virtuais utiliza geometria poligonal, combinando formas básicas, texturas e cores.

Atualmente compõem a coleção de mundos virtuais os seguintes tópicos:

1. NewtonWorld: experimentos virtuais relativos às Leis de Newton, que podem ser utilizados para exploração dos conceitos de inércia, conservação da energia cinética e conservação da quantidade de movimento. Os estudantes acessam o experimento virtual e podem simular alterações dos objetos presentes, através de menus, nos experimentos de forma a visualizar o comportamento de diferentes massas e diferentes colisões. Nos experimentos, sensações relativas à mudanças de sombras representam as diferentes variáveis da física.

As Figuras 3.15 e 3.16 são relativas ao experimento de conservação da energia e da quantidade de movimento. Pode-se observar que há um corredor onde corpos deslizam e dois muros ao final do dito corredor.
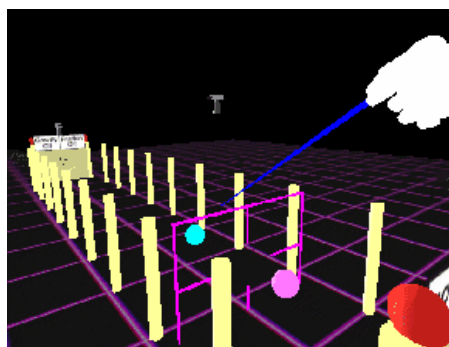
Fig. 3.15 - AV de um Experimento - NewtonWorld
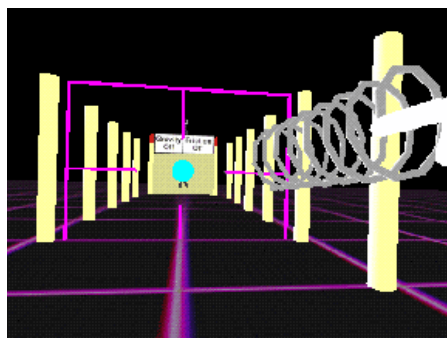
Fig. 3.16 - Simulando colisão no NewtonWorld

2. *MaxwellWorld*: Conjunto de mundos virtuais elaborados para prover a examinação e aprendizado das forças e campos eletrostáticos, capazes de permitirem que o estudante descubra empiricamente as Leis de Gauss. A Figura 3.17 refere-se a um destes mundos virtuais, desenvolvido para visualização de campos elétricos de uma carga puntiforme.
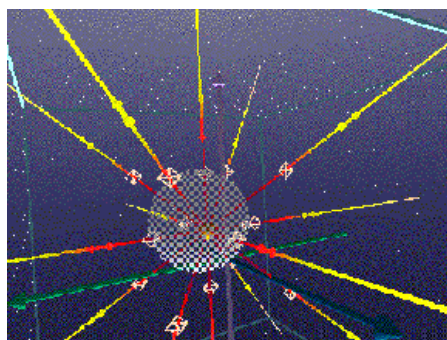
Fig. 3.17 - Campo Elétrico de uma Carga Puntiforme - MawellWorld

3. *PaulingWorld:* São os componentes mais recentes do projeto e visam auxiliar professores e alunos, de forma a ser uma ferramenta de pesquisa. É composto de mundos virtuais que permitem examinar a estrutura de moléculas simples e compostas. A Figura 3.18 apresenta um exemplo de mundo virtual relativo à representação molecular.
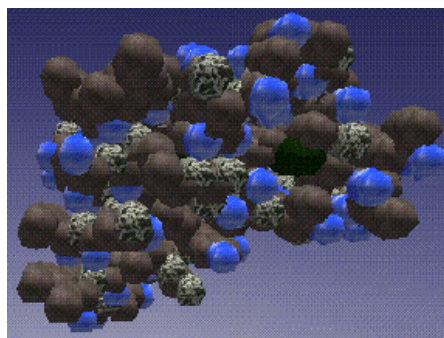
Fig. 3.18 - Mundo Virtual relativa à uma Molécula

3.3.2.1. Vantagens do Sistema

Destacam-se como principais vantagens do sistema:

(a) os experimentos virtuais desenvolvidos no ScienceSpace possibilitam efetiva imersão do usuário nos mundos virtuais com uma boa utilização dos periféricos adequados;

(b) em função da larga experiência no desenvolvimento e nas diversas realimentações obtidas pelos autores ao longo destes anos, um grande conjunto de melhorias tem tornado o aprendizado através dos mundos virtuais mais efetivo e permitido melhorias na motivação dos estudantes com respeito ao interesse pelos tópicos abordados (Dede et al., 2000);

(c) o aprimoramento dos mundos virtuais tem tornado cada um deles mais adequado aos diferentes perfis de estudantes, o que diferencia-os em termos de adequabilidade;

3.3.2.2. Desvantagens do Sistema

As principais desvantagens do sistema são:

(a) impossibilidade de aprendizado colaborativo: o sistema não permite que diferentes estudantes, em pontos remotos ou em grupo, possam, simultaneamente, partilhar os mesmos ambientes virtuais;

(b) custo do equipamento envolvido: estações de trabalho *Silicon* e os periféricos utilizados são muito caros;

(c) mundos virtuais previamente definidos pelos autores do projeto: embora tais definições sejam o resultado de muitos anos de investigação, os usuários não podem

criar novas simulações ou promover alterações, que não sejam paramétricas, no mundo virtual em utilização;

(d) o sistema não partilha o ambiente simulado com usuários desprovidos do periférico adequado: não é possível que um segundo aluno, que não esteja com um capacete, possa observar a simulação que o aluno que utiliza o sistema momentaneamente está executando;

(e) há um número limitado de simulações;

(f) impossibilidade de extensão dos mundos virtuais para os computadores pessoais (PC´s).

3.3.2.3. Análise das Contribuições do Sistema

Uma análise do *ScienceSpace* e de sua utilização apresenta importantes conclusões, relativas à utilização de mundos virtuais na educação, a saber:

i) estudantes apresentam grandes diferenças com respeito ao estilo de interação com os mundos virtuais, incluindo as habilidades para manuseio de ambientes 3D e a capacidade para utilizar os periféricos sem mal estar;

ii) a imersão não dispensa outras formas de interação, assim, estudantes que estão utilizando um capacete e estão imersos em um experimento virtual são pouco capazes de acessar instruções escritas ou completar testes (escritos). Neste caso em particular, interações verbais são mais adequadas;

iii) os capacetes podem causar desconforto para os usuários, o que leva a crer que em algumas situações, a imersão subjetiva pode ser mais adequada, embora esteja relacionada com perda parcial da imersão;

iv) o aprendizado é melhorado com repetidas situações de experimentos, assim, o aluno consegue reter mais conteúdo com a repetição da experimentação do que com um único contato com o conteúdo. Tal conclusão reforça argumentos da necessidade de experimentos virtuais disponíveis para reuso em outro tempo e local;

v) uma nova forma de entrega do conteúdo, suportado por RV é muito mais motivante e cada alteração provida nos experimentos relativa à inserção de novos recursos de interação (tais como a adição de uma imagem estéreo ou a inclusão de um atuador de força) provê ainda mais motivação;

vi) um time de desenvolvimento multidisciplinar (engenheiros, psicólogos, cientistas da computação, pedagogos e estudantes) provê condições de melhorias constantes no projeto.

3.3.3. ATSWorlds - Usando RV para modelagem de objetos 3D

ATSWorlds é um sistema desenvolvido na Unisinos - Universidade do Vale do Rio dos Sinos, com a finalidade de criar objetos e mundos 3D modelados através de diferentes operações de *sweep* com uso de VRML e Java (Silva; Osório, 2000)

Sweep é uma técnica que permite criar sólidos de revolução. Através de um contorno descrito por linhas em 2D realiza-se um deslocamento e à medida que este contorno descreve uma trajetória no espaço 3D, sólidos são gerados de forma a definir um objeto, sendo considerada uma maneira natural e intuitiva de construção.

O sistema ATSWorlds permite que um usuário, usando um *applet* Java, defina a linha de modelagem e visualize o objeto final em 3D em uma janela, dentro da mesma página, com uso de VRML. O *applet* permite alterar a definição da linha e coordena a visualização de novas formas na janela de apresentação do objeto gerado por *sweep*, como pode ser visto na Figura 3.19.

O sistema trabalha com a hipótese de visualização de formas, permitindo, com utilização de páginas da WEB, a verificação da técnica de *sweep*. Apresenta como principais vantagens:

i) visualização de formas 3D geradas dinamicamente, com rápida resposta à solicitação do usuário;

ii) uso na WEB, de forma gratuita, tornando-se um sistema portátil e compatível com a maioria dos sistemas operacionais (exige-se que o navegador esteja associado a um *plug-in* compatível com Java e outro *plug-in* compatível com VRML).

O sistema tem como principais limitações:

i) impossibilidade de dotar as formas de movimentos e animações, apresentando resultados estáticos e que permitem exclusivamente a interação na forma de visualização;
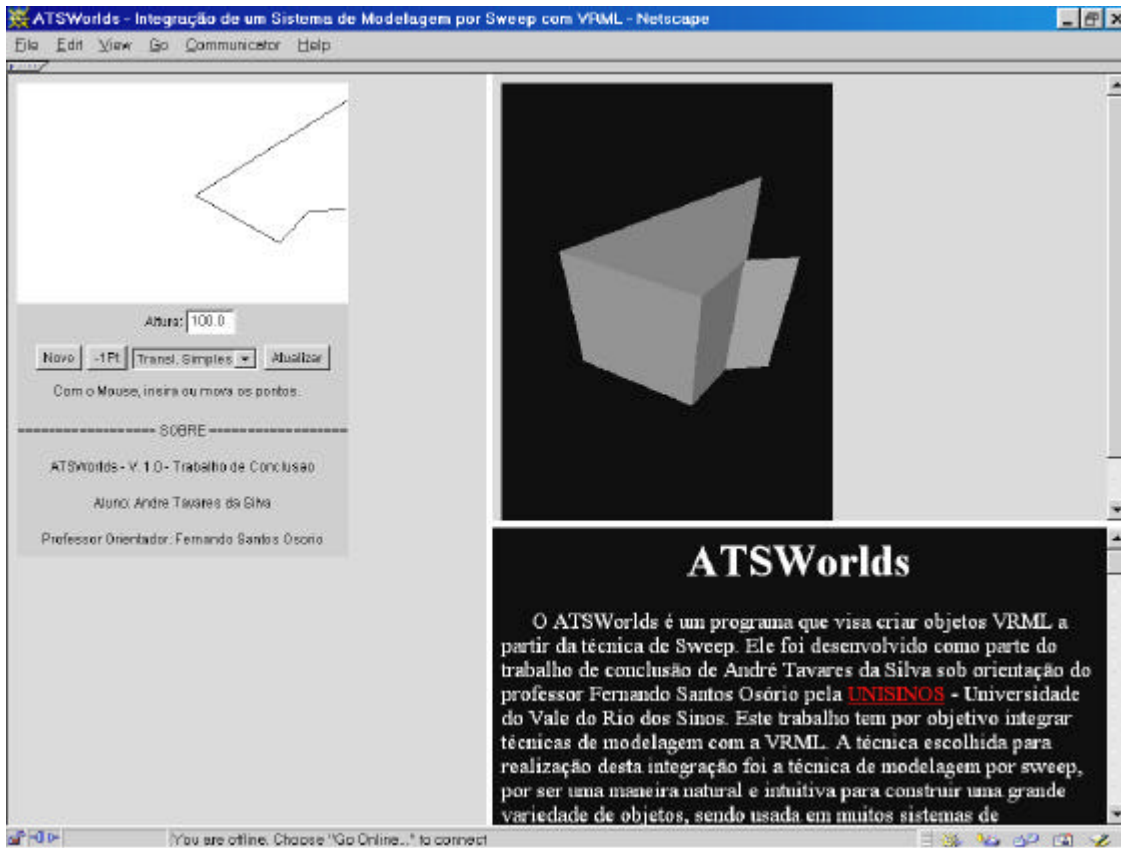
Fig. 3.19 - Exemplo de Geração de Polígonos no ATSWorld

ii) não permite que o usuário, se estiver usando a versão compatível com a WEB, salve seu trabalho e possa reutilizá-lo, se necessário;

iii) não permite definições de atributos de cor, transparência, reflexão e textura dos objetos gerados;

iv) impossibilita a criação de cenas com mais de um objeto.

3.3.4. AVIT - Ambiente Virtual Interativo

Pesquisadores da Universidade Federal de São Carlos têm trabalhado na criação de um ambiente virtual interativo, utilizando Java e VRML para concepção de uma plataforma para desenvolvimento de aplicações interativas de Realidade Virtual, o AVIT - Ambiente Virtual Interativo (Ipolito; Kirner, 2000).

AVIT é um sistema centralizado de RV multi-usuário que, através da combinação VRML/Java possui dois módulos: o módulo servidor (construído em Java) e o módulo cliente (construído em VRML e Java). Com o uso de um navegador de WEB compatível e o módulo cliente, um usuário pode escolher um avatar, entrar com seus dados e interagir com outros usuários conectados simultaneamente. O usuário, quando entra no sistema, adentra num mundo virtual 3D que pode ou não ser uma representação de um cenário real, como um museu virtual.

Podem ser experimentados um barzinho virtual, uma visita a um museu de arte moderna, uma transação bancária em um banco virtual, assistir a um filme, passear por um parque entre outros.

Com o aumento de pessoas que freqüentam determinado mundo virtual pode surgir uma comunidade virtual distribuída, ou seja, o conjunto de pessoas que interagem em um ambiente virtual distribuído. As pessoas podem simplesmente se divertir, discutir idéias ou até mesmo participar de aulas em laboratórios virtuais comandadas por professores virtuais.

A Figura 3.20 representa uma das possibilidades de uso do AVIT, através da navegação em um museu virtual, onde o usuário encontraria obras como a Mona Lisa e a Santa Ceia.

Observa-se uma janela de apresentação de um mundo virtual simulando um museu ao ar livre - concebida em VRML, uma área de bate papo - *chat* - com a possibilidade de envio e recebimento de mensagem - desenvolvida sobre um *applet* Java, alguns objetos que simulam *avatar* e descrição do usuário e a seleção de conexão com o servidor.

Uma segunda pasta apresenta usuários conectados e suas características. A utilização inicia-se com a escolha de uma representação - *avatar* - que simule o usuário e que será inserido no cenário virtual. Após escolha, o usuário solicita conexão e passa a navegar simultaneamente com outros usuários, passando a interagir com o mundo.

Fig. 3.20 - Página WEB do Programa AVIT - Página Inicial

A implementação foi desenvolvida com utilização de VRML e Java: VRML para simulações do ambiente 3D e do *avatar* e Java contribui com *applets* que coordenam a conexão com o servidor, interação entre usuários e entre usuário/mundo virtual, além de ser fundamental no módulo servidor.

O projeto demonstra a viabilidade de união Java/VRML para elaboração de ambientes virtuais colaborativos, na Internet, que possibilitem diversos usuários trocarem informações e interagirem mediante um mundo virtual, sem, no entanto permitir que usuários modifiquem o que está sendo visto ou que criem seus próprios mundos virtuais.

3.3.5. GEO 3D - RV no Ensino da Geometria Espacial

GEO 3D é um software educacional implementado em um ambiente de realidade virtual para ensino de geometria espacial, desenvolvido pelo grupo de Pesquisa de Realidade Virtual do departamento de informática da UNISC (Keller; Schreiber, 1999).

O objetivo básico do programa é possibilitar um processo de aprendizagem de tópicos de geometria espacial através da interação, exploração, descoberta e observação de sólidos geométricos.

O software foi desenvolvido para ser usado em rede através de navegadores compatíveis com JavaScript e VRML. Usando a integração de HTML, JavaScript e VRML apresenta páginas interativas onde aprendizes podem navegar e testar seus conhecimentos de aspectos da geometria espacial.

HTML e JavaScript prestam-se a operacionalizar o ambiente para a WEB e VRML para geração de figuras tridimensionais que também comporão as páginas, possibilitando navegação e visualização.

A metodologia do ensino foi baseada em sistemas do tipo CAI, em que o computador é usado para controlar e acompanhar a apresentação de determinada quantidade de informações ao aluno, dispondo de interação e na qual solicitações do computador são respondidas pelo aluno e, a partir das respostas, novas informações são fornecidas.

Para este fim, a ferramenta disponibiliza:

(a) Tutorial: apresentação do conteúdo teórico ao aluno, como a teoria relativa a poliedros, que é apresentada na Figura 3.21;

Fig. 3.21 - Página de Teoria de Poliedros do Programa GEO 3D.

(b) Exercício e Prática: uma série de exercícios para teste de conhecimentos, com respostas e uma breve estatística, como a página apresentada na Figura 3.22, em que um exercício de poliedros é apresentado ao usuário;
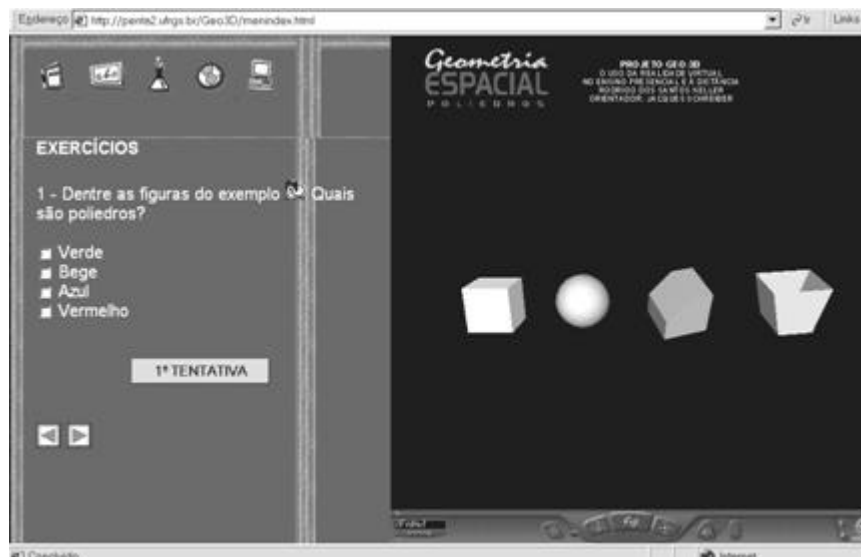
Fig. 3.22 - Página de Exercícios no GEO3D

(c) Simulação: objetos geométricos reais são simulados em arquivos VRML, permitindo a manipulação;

(d) Jogo Educacional: um jogo permite ao aluno testar seus conhecimentos de forma motivadora. A Figura 3.23 retrata o ambiente de Jogo Educacional do sistema.

Fig. 3.23 - Página de Jogo Educacional do GEO3D

A utilização do programa se dá com o acesso ao sítio do mesmo na Internet, onde se encontra a página inicial. Após selecionar "Iniciar", o usuário encontra uma segunda tela onde pode começar a usar o programa para aprender e exercitar conhecimentos de geometria espacial.

Destacam-se como principais diferenciais do sistema:

(a) O desenvolvimento de tópicos de Geometria, disciplina que geralmente ocasiona grande dificuldade de visualização aos alunos em uma ferramenta gratuita compatível com a Internet, de forma que a mesma é suportado pelos navegadores convencionais e pelos mais diversos sistemas operacionais;

(b) a forma de concepção dos ambientes virtuais e da entrega do aprendizado fundamentada em métodos pedagógicos, o que propicia a entrega do conteúdo de forma estruturada.

A aplicação descrita está fortemente relacionada ao domínio de Geometria Espacial, e tem como principais limitações:

(a) Flexibilidade: a impossibilidade de um usuário professor definir, a seu critério, aulas interativas, que possam ser usadas em acordo com uma necessidade específica;

(b) Apresentação estática, com mundos virtuais previamente definidos e com pouco dinamismo de formas;

(c) Não há interação com os objetos do mundo virtual.

3.3.6. LVEE - Laboratório Virtual para Experiências de Eletrônica

Desenvolvido na Pós-Graduação da Pontifícia Universidade Católica de Campinas (PUC Campinas), o LVEE é um ambiente virtual que permite a estudantes efetuarem experimentos virtuais de eletrônica, com objetivo de solidificar a teoria de circuitos eletrônicos. Desenvolvido com uso de VRML e a *EAI (External Authoring Interface)* com Java, caracteriza-se pela troca de informações entre um *applet* e um mundo virtual (Meiguins; Behrens, 1999)

O projeto se fundamenta na concepção de um ambiente virtual que adota modelos do mundo real usados nas disciplinas de Circuitos Elétricos, Eletrônica Básica, Circuitos Lógicos e Arquitetura de Computadores. Experiências com utilização de resistores, capacitores, indutores, amperímetros, voltímetros e fontes são disponibilizadas por modelos

destes elementos simulados em VRML e cujos parâmetros e disposição no circuito são definidas por um *applet* Java. Assim, um usuário que deseje uma simulação deve selecionar componentes, dispô-los em circuito e definir ou modificar seus parâmetros, como pode ser visto na Figura 3.24.
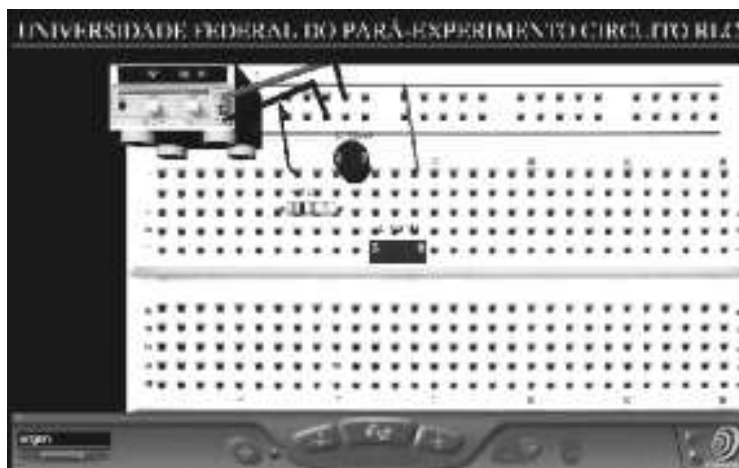
Fig. 3.24 - Desenvolvendo Experiência no LVEE

Após definição, o programa aciona o software PSPICE® - *Simulations Program with Integrated Circuit Emphasis for PC* - que faz os cálculos de variáveis do circuito - tensão e corrente e gera um arquivo de saída. Tal arquivo é lido pelo *applet* Java que atualiza os ambiente virtual com as informações recebidas. O *applet* também recebe informações de modificações de parâmetros e aciona o programa PSPICE® para novas simulações, se necessário.

O sistema permite que notas de aula sejam mostradas em um quadro que simula quadros de aula e a apresentação de um vídeo em uma TV que está situada no laboratório virtual.

As principais limitações do programa são:

(a) não armazena as experiências efetuadas pelos alunos para uma posterior avaliação, que poderia ser efetuada de forma pedagógica junto ao professor;

(b) limitação dos componentes passivos a resistores, capacitores e indutores;

(c) as fontes, voltímetros, amperímetros, osciloscópios e gerador-de-funções são elementos meramente figurativos no mundo, não tendo nenhuma funcionalidade, com

exceção da fonte, que apresentará somente o valor de sua excitação senoidal, sem possibilidade de utilizar seus botões;

(d) limitações da apresentação dos componentes no mundo virtual, tendo somente duas posições de cada componente, como por exemplo diagonal, horizontal e vertical.

3.3.7. Physicsweb - Laboratório Virtual de Física

Physicsweb (2000) refere-se com um sítio da Internet que disponibiliza diversas animações na forma de *applet*, VRML e Schockwave, definidas por pesquisadores de todo o mundo e disponibilizadas gratuitamente.

É grande o número de *applets* Java em diversos links oferecidos pelo sítio, onde podem ser obtidos visualizações 2D de conceitos como reflexão e refração da luz, experiência de Young (difração da luz), entre outros.

Um das seções do sítio, mais relacionada aos objetivos deste trabalho, contém ambientes virtuais concebidos em VRML para apresentação de conceitos de Física, de forma a permitir ao usuário interação e modificação de parâmetros físicos e os efeitos de tais modificações.

Para relato, separou-se a experiência de um pêndulo imerso ao mesmo tempo em um campo magnético e em um campo gravitacional, que pode ser visualizado na Figura 3.25. O pêndulo sofre modificações de comportamento na oscilação de acordo com a intensidade dos campos envolvidos que são definidas pelo usuário através de toques nos botões de controle que se encontram abaixo do pêndulo. O usuário tem a sensação de observar e participar de um experimento real, com a vantagem de estar visualizando linhas de campo magnético - que são imaginárias abaixo do pêndulo.

As principais limitações dos programas apresentados pelo sítio - limitações comuns a todos eles são:

(a) impossibilidade do usuário manipular os objetos inserindo-os e retirando-os, de forma a experimentar variações nos mundos virtuais apresentados;

(b) não permitem que o usuário crie seus próprios mundos virtuais e armazene-os para trabalhos futuros;

(c) inserção de textos explicativos no meio dos cenários virtuais, levando a confusão do usuário durante utilização.
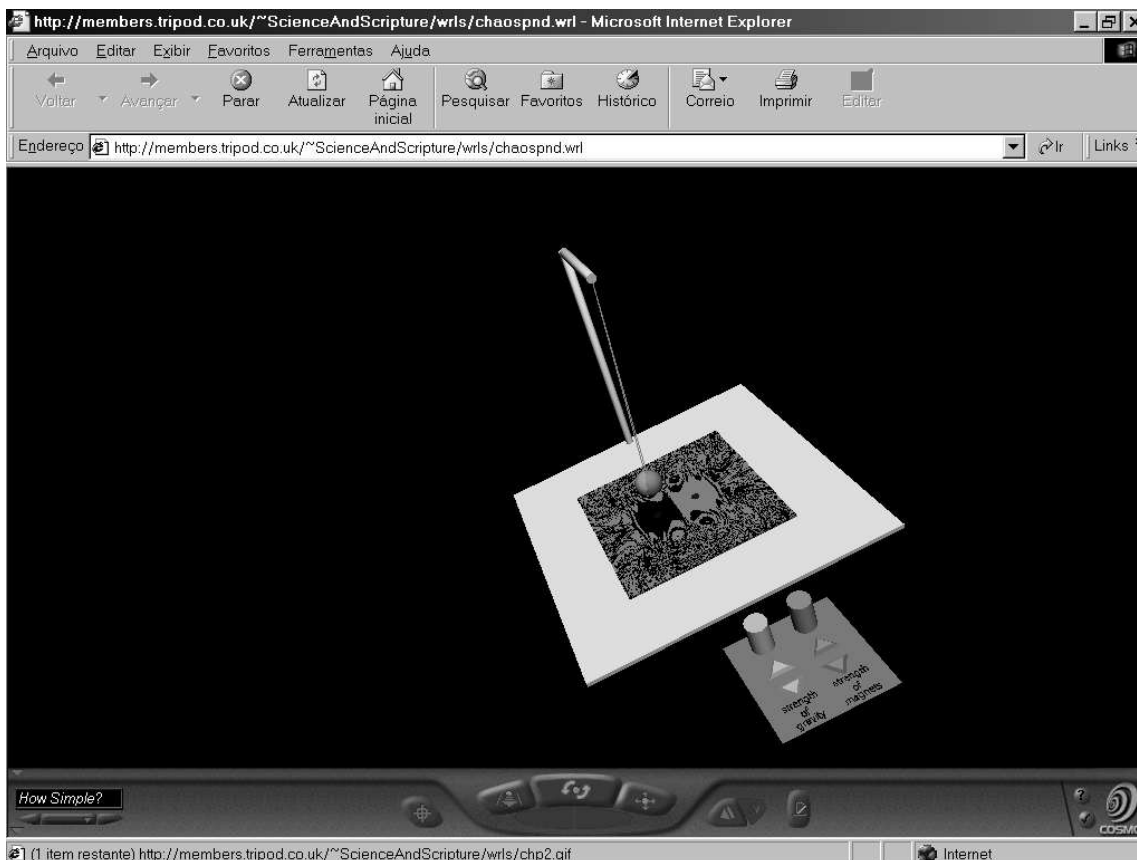
Fig. 3.25 - Pêndulo sujeito a Campos Magnético e Gravitacional.

3.3.8. VRML Gallery of Electromagnetism

Trata-se de um sítio da Internet (Salgado, 2001) que disponibiliza um conjunto de mundos estáticos desenvolvidos em VRML relacionados ao conteúdo de Física, principalmente com Eletromagnetismo. A galeria de experimentos apresenta diversas simulações, tais como: adição de vetores, produtos vetoriais, campos elétricos de cargas puntiformes, dipolos elétricos e esferas condutoras, campos magnéticos gerados por correntes elétricas (Lei de Ámpere) e outros relacionados a fundamentos da eletricidade.

As animações em VRML permitem aos usuários navegar por espaços 3D onde se encontram cargas elétricas, cor rentes elétricas e vetores relacionados com Campos Elétricos e Magnéticos. A Figura 3.26 apresenta a visualização disponível para a Lei de Coulomb, onde vetores Campo Elétrico de uma carga puntiforme são apresentados.
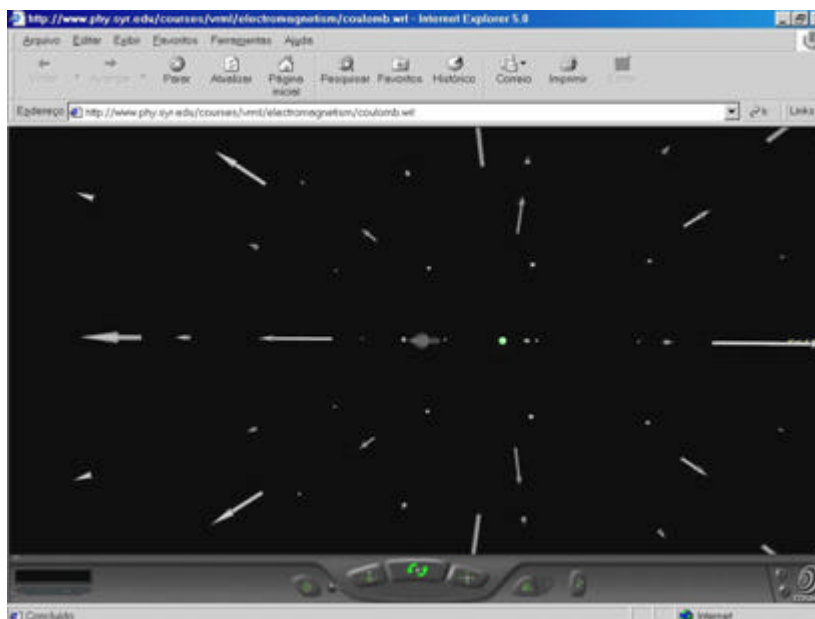
Fig. 3.26 - Lei de Coulomb por *VRML Gallery of Eletromagnetism*

Embora existam diversas simulações, as mesmas são representações estáticas de conceitos da eletricidade e não permitem ao usuário modificar ou alterar parâmetros de forma a percepção de alterações de características dos Campos (Elétrico/Magnético) por alteração de parâmetros dos elementos fonte (Cargas elétricas estáticas ou em movimento).

O sítio disponibiliza ainda alguns *gif* animados que ilustram os conceitos relacionados com a lei de Ámpere e integral de linha.

3.3.9. WebTOP - Sistema de Experimentos de Óptica Geométrica na Web

O Projeto Óptica (The Optics Project) - TOP é um sistema de computação gráfica tridimensional interativo para visualização de fenômenos ópticos, visando ensino de alunos de curso de graduação de universidades americanas.

Foi iniciado na *Missisipi State University* no Centro de Pesquisas para Simulações Computacionais em 1994, motivado pelo interesse de conceber lições de óptica geométrica para alunos de graduação. Inicialmente projetado para ser executado em estações de trabalho da *Silicon Graphics*, foi implementado com o uso de *Open Inventor*. O sistema WebTOP é a versão para a Web do sistema TOP (Kiril et al., 2000), incorporando as características originais de TOP e ainda as vantagens de soluções para a Web, tais como

portabilidade, integração simples com textos e imagens e capacidade de colaboração com outros usuários. .

WebTOP é um sistema interativo com simulações de óptica geométrica apresentadas em cenários tridimensionais, implementado com utilização de VRML, Java e a *External Authoring Interface* - *EAI* (Marrin, 2001). WebTOP pode ser executado em diversas configurações de sistemas operacionais/navegadores/hardware. Trata-se do primeiro sistema interativo e tridimensional completo de óptica.

WebTOP consiste de cinco sub-módulos de três diferentes áreas da óptica, relacionadas com fenômeno da difração e polarização da Luz (fenômenos ondulatórios que alunos têm dificuldade de visualizar ou abstrair).

A interface com o usuário apresenta cinco partes básicas: o Menu de Módulos, a Cena, o Painel do *CosmoPlayer* (CosmoPlayer, 2000), o Console e o Menu de Atividades, como pode ser visto na Figura 3.27.
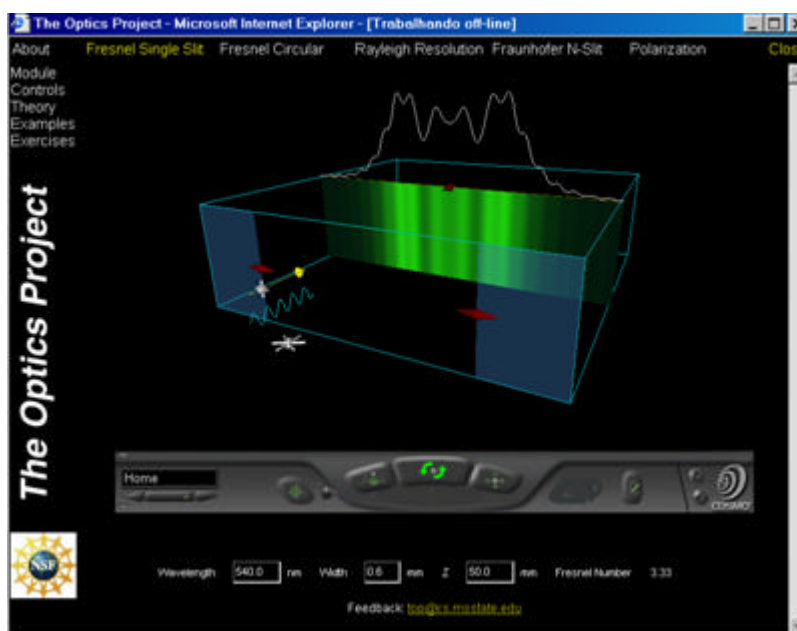
Fig. 3.27 - Página Típica do Sistema WebTOP

O Menu de Módulos permite que o usuário, através da seleção com o mouse, acesse uma das simulações de fenômenos ópticos disponíveis, evitando que o usuário seja obrigado a voltar a uma página que contenha links para todas as simulações.

A Cena contém a representação tridimensional do fenômeno óptico utilizando VRML para implementação e ocupando maior parte da página Web. Usualmente, é constituída de uma grande variedade de elementos da óptica que podem ter seus parâmetros modificados por manipulação direta. Quando o usuário passa com o cursor sobre elementos da cena que constituem uma *widget*, um texto explica o que pode ser manipulado com esta *widget*.

O Console corresponde a um *applet* Java que contém alguns parâmetros que podem ser modificados pela entrada de dados feita pelo usuário através do teclado, além de informar os valores atuais dos parâmetros do experimento em curso.

O Menu de Atividades lista as cinco atividades básicas disponíveis:

(a) Módulos: atividades relacionadas com exibição de simulações;

(b) Controles: apresenta outras páginas Web que explicam como usar o navegador VRML e como mudar valores de parâmetros dos elementos de cena. Trata-se de um conjunto de textos destinados a novos usuários;

(c) Teoria: ativa seqüência de páginas Web que contêm a teoria relativa ao módulo corrente. Contém um conjunto relevante de notas, essencial para o aprendizado teórico correspondente ao módulo;

(d) Exemplos: leva a um documento HTML que é uma lista das sessões WebTOP previamente definidas e demonstram bem o que pode ser feito com os módulos;

(e) Exercícios: ativa um documento HTML que contém uma lista de exercícios, forçando o usuário a interagir com o programa de forma a encontrar respostas aos quesitos propostos.

A implementação dos módulos levou em conta a separação de geometria/comportamento dos elementos. A geometria estática é armazenada em mundos definidos em VRML, enquanto que o comportamento é definido pela utilização de Java que coordena os elementos da cena através da *EAI*.

Há dois tipos básicos de objetos de interface (*widgets*):

- *Widgets* 3D que permitem manipulação direta de objetos existentes na cena e provêm uma interface natural e intuitiva de controle de parâmetros que são entrada para o sistema físico apresentado;

- *Widgets* 2D que estão definidas com o uso de Java e constituem o Console apresentando diversos parâmetros da simulação, provendo controle adicional destes parâmetros.

3.3.9.1. Utilização do Sistema WebTOP

A utilização do sistema se dará, por exemplo, com a visualização da solução proposta para a Difração da luz (*Frounhofer Diffraction*), supondo que um feixe de luz incida em um anteparo, após sofrer difração por fendas. Para este fim, o usuário pode, através do Console:

- variar o número de fendas - *slits*;
- variar o comprimento de onda que está sendo analisado - *wavelength*;
- alterar a largura das fendas - *width* ;
- modificar a distância entre as fendas - *distance;*

Os parâmetros acima podem ainda serem modificados com as *widgets* 3D que se encontram na Cena, através de cliques e arrastes. A alteração de parâmetros faz com que o resultado da nova simulação seja imediatamente apresentado, com a variação das franjas de interferência.

A Figura 3.28 apresenta dois momentos da simulação, onde o usuário altera as larguras das três fendas, para a segunda cena.

A Figura 3.29 apresenta os resultados da variação do número de fendas, a princípio, com uma fenda e numa segunda cena, a simulação com duas fendas.
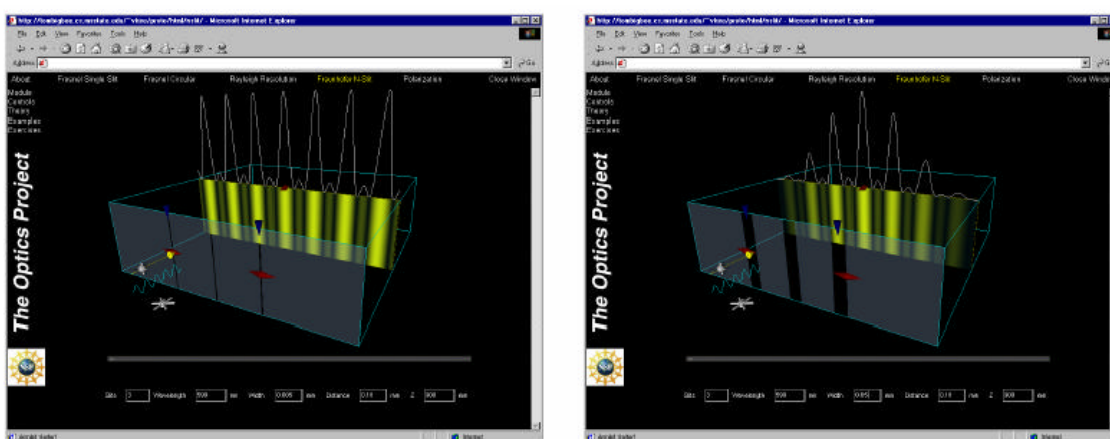
Fig. 3.28 - Simulação da Difração da Luz - Variação da Largura das Fendas
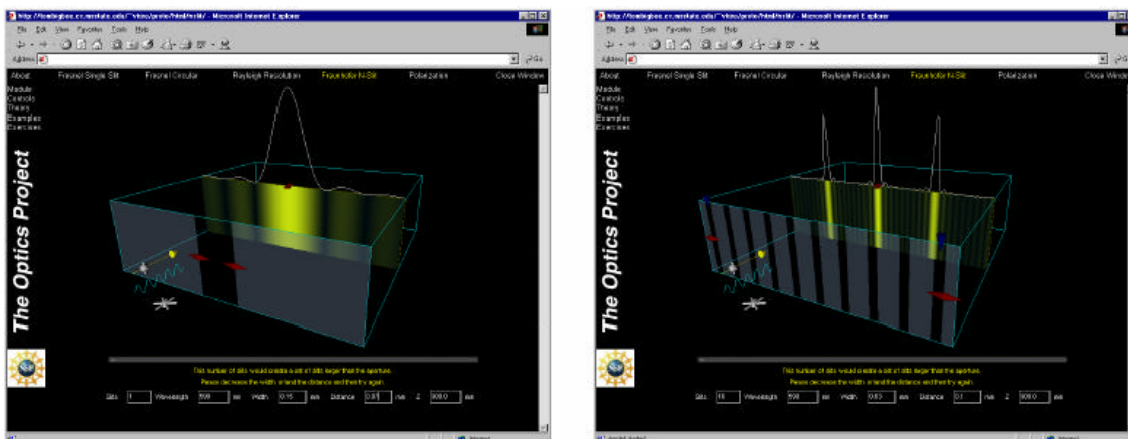
Fig. 3.29 - Simulação da Difração da Luz, com Uma Fenda e com Dez Fendas

3.3.9.2. Resultados de Utilização

Segundo Kiril et al. (2000), TOP e WebTOP tem sido usado como ferramenta de auxílio a cursos de Óptica para alunos de graduação em três diferentes universidades americanas: (a) Mississipi State University (MSU), (b) University of North Carolina e (c) North Carolina A&T State University.

Os sistemas foram utilizados de três maneiras distintas. Primeiro, para apresentar conteúdo de experimentos em uma aula normal. Segundo, solicitou-se dos estudantes a solução de problemas em casa, usando os sistemas. Terceiro, os sistemas passaram a fazer parte dos projetos dos estudantes, que tiveram de desenvolver um dado tema usando os sistemas.

A utilização tem dado bons resultados na motivação dos estudantes, na demonstração de conceitos por professores em laboratórios computacionais e tem permitido que estudantes possam explorar conceitos de óptica geométrica em suas casas.

3.3.9.3. Limitações do Sistema

WebTOP apresenta um conjunto limitado de simulações de experimentos envolvendo conceitos de Óptica, totalizando cinco simulações diferentes relacionadas a dois tópicos principais de ondulatória.

O sistema tem como principais limitações:

(a) Não permite que usuários professores possam criar ou modificar os módulos existentes, particularizando-os;

(b) As atividades permitidas pelo sistema também não podem ser alteradas pelo professor, que, portanto não pode modificar os textos explicativos, exercícios ou exemplos relacionados com os módulos, sendo a utilização restrita a modificação de parâmetros;

(c) Impossibilidade de inserção de novos objetos nas cenas previamente definidas.

3.5. Análise Comparativa

A leitura das soluções destacadas neste capítulo permite evidenciar algumas características relevantes à concepção de sistemas destinados a Educação Virtual Interativa, tais como os laboratórios virtuais, enumeradas a seguir:

i) utilização de ambientes 3D para concepção dos experimentos virtuais permitem a interação mais natural e realística. A utilização de diagramas 2D está mais próxima do que é apresentado em livros que do contexto de um laboratório virtual (Kiril et al., 2000);

ii) os usuários preferem manusear experimentos concebidos por si mesmo que apenas visualizar ou interagir com experimentos pré-definidos. A utilização de soluções pré concebidas acaba por se tornar um fator desmotivador para as comunidades de ensino/aprendizado que utilizem tais estratégias (Hiltz; Wellman, 1997);

iii) sistemas capazes de absorverem novos componentes e permitirem que os usuários armazenem suas alterações para posterior reutilização são mais extensíveis, provendo maior possibilidade de concepção de novos experimentos;

iv) Johnson (1998) salienta que é muito importante que o desenvolvimento de tais soluções esteja vinculado a um dado domínio de aplicação, sob pena de perda dos objetivos e da concepção de soluções pouco contextualizadas;

v) a solução apresentada em programas como o Alice (Pierce et al., 1997), ou o WebTOP, reforça que a utilização de interfaces 3D interativas, que dispensam o aprendizado de tópicos de Computação Gráfica (tais como transformações gráficas, modelagem geométrica etc) facilitam a utilização por um grupo maior de usuários e devem ser fator de diferenciação na proposição de sistemas (Green; Halliday, 1996);

vi) a concepção de sistemas que são compatíveis com a Web aumenta muito a possibilidade de utilização dos mesmos de forma assíncrona pelos estudantes e professores, que, de qualquer terminal de computador podem fazer utilização dos mesmos, tornando tais soluções muito atrativas (Kiril et al., 2000, Lawhead et al., 1997).

Considerando-se as limitações apresentadas pelos sistemas analisados no capítulo três e as características apontadas, obtém-se a Tabela 3.1, que, apresenta esquematicamente, uma comparação das soluções em destaque neste capítulo. A tabela compara-as por meio de itens importantes na definição de projeto dos sistemas, dando destaque às diferenças relacionadas aos propósitos deste trabalho. As linhas contém os trabalhos relacionados e as colunas, os itens comparados. Itens hachurados indicam uma característica válida para o elemento em comparação e ausência de hachuras apontam uma característica inexistente para o sistema listado.

Como pode ser observado, a tabela não indica a impossibilidade, mas a inexistência de um sistema que atenda simultaneamente a todas as características realçadas, o que motivou a especificação de uma solução que contemple tais propósitos simultaneamente. Por esta razão, o próximo capítulo analisa e propõe o projeto de um sistema computacional para atender simultaneamente todos os requisitos destacados nas linhas da Tabela 3.1.

Tabela 3.1 - Comparação entre os Sistemas Apresentados

	ALICE	AVIT	ATSWorld	Gallery of Eletrom.	Geo3D	Labor. Virtuais (Java)	LVEE	PhysicsWeb	ScienceSpace	WebTop
Acessível via Web	■	■	■	■	■	■	■	■		■
Armazena Alterações dos Usuários	■									
Arquivos de Exemplos	■	■	■	■	■		■	■	■	
Associação com um dado Domínio de Conhecimento			■	■	■	■	■	■	■	■
Compatibilidade com Diversos Sistemas Operacionais		■	■	■	■	■	■	■		■
Definição de Novos Ambientes/ Experimentos	■					■	■			
Dispensa o Aprendizado de alguma Forma de Programação		■	■	■	■	■	■	■	■	■
Dispensa prévio conhecimento de Computação Gráfica	■	■		■	■	■	■	■	■	■
Inserção de novos objetos em um dado Cenário/Experimento	■					■				
Simulações com uso de Ambientes 3D	■	■	■	■	■		■	■	■	■

4. Especificação do Sistema

A partir da análise comparativa de sistemas similares apontada no final do capítulo três, este capítulo apresenta a proposição de um sistema capaz de atender simultaneamente as características realçadas como ausentes nos sistemas similares, destacadas na Tabela 3.1.

Tal proposição dar-se-á por meio da análise e projeto do sistema, alvo desta tese, destacando a arquitetura e os detalhes estruturais do mesmo. São relatados os passos para a definição do software, aspectos particulares da análise e do projeto e a descrição dos componentes. Para tais propósitos, a proposta do sistema foi desenvolvida com análise orientada a objetos e com uso da UML (*Unified Modeling Language*), que representa uma ferramenta capaz de modelar qualquer tipo de aplicação de tempo real, cliente/servidor ou outros tipos de software (Furlan, 1998).

4.1. Análise do Sistema

4.1.1. Escopo e Propósitos

Para possibilitar a concepção de experimentos virtuais de Física, que possam ser utilizados como ferramentas de ensino, o sistema desenvolvido é compatível com computadores pessoais - PC, que, devem estar conectados à rede mundial de computadores - Internet. Tais computadores podem utilizar diferentes tipos de sistemas operacionais, desde que os mesmos possibilitem a utilização dos navegadores mais usuais da rede.

Considerando-se as limitações dos sistemas destacadas no capítulo três e realçadas graficamente na Tabela 3.1, e com o propósito de desenvolver uma solução que preencha todas as características listadas simultaneamente, o sistema proposto por esta tese visa:

(a) propiciar o desenvolvimento de experimentos virtuais de Física, relacionados com Mecânica Clássica, Óptica e Eletricidade e adequados ao domínio do Ensino Médio, para utilização de tais experimentos por professores e alunos;

(b) possuir uma interface amigável, de fácil aprendizado que permita a utilização por não programadores no desenvolvimento de experimentos virtuais de Física, utilizando técnicas de Realidade Virtual. Para tanto, disporá de uma biblioteca de elementos, provendo condições de concepção de cenários com objetos que comportar-se-ão de acordo com as leis da Física, abstraindo o comportamento real;

(c) armazenar os experimentos desenvolvidos, de forma a prover reutilização de código e permitir a difusão destes experimentos, tornando-os acessíveis a usuários da WEB;

(d) privilegiar ferramentas e linguagens de programação abertas e ser independente do sistema operacional;

(e) permitir a incorporação de novos objetos que, porventura, venham a ser definidos, com a inserção dos mesmos na biblioteca de objetos, provendo utilização para a construção de experimentos de Física, desde que a definição esteja em acordo com os modelos de objetos que fundamentam o sistema;

(f) permitir a realização de experimentos virtuais de Física, de forma não imersiva, em 1ª pessoa, em cenários virtuais relacionados, propiciando condições de melhoria do estudo/aprendizado individual de forma motivadora.

Os propósitos do sistema serão avaliados por diferentes clientes do sistema (alunos e professores) que, desprovidos de conhecimentos técnicos de Computação Gráfica e Realidade Virtual consigam desenvolver experimentos virtuais de Física através do sistema, sem a necessidade da utilização de ambientes virtuais previamente concebidos.

4.1.2. Restrições

São restrições do sistema:

(a) permitir desenvolvimento e visualização de experimentos de Física no domínio relativo a todo conteúdo do ensino médio (2º grau), não suportando, por exemplo, experimentos relacionadas à relatividade;

(b) exigir a definição dos relacionamentos entre elementos de cena, não provendo associações automáticas entre elementos e destes com sensores ou atuadores.

4.1.3. Metodologia de Desenvolvimento

Com o propósito de desenvolvimento de soluções de software com uso de Realidade Virtual (RV), a pesquisa de alternativas à metodologia aponta a necessidade de considerar que os objetos do domínio do problema devem ser modelados considerando-se aspectos relativos à geometria e ao comportamento (interação entre objetos de cena e com objetos de interface), caracterizando-se duas modelagens importantes: a modelagem geométrica e a modelagem comportamental.

A modelagem geométrica visa definir atributos relacionados à geometria dos objetos, tais como: dimensões, posição no espaço, cor, textura etc. Já a modelagem comportamental diz respeito à definição do comportamento do objeto e de suas relações com outros objetos. Neste aspecto, são definidas características como massa, velocidade, coeficiente de deformação elástica e temperatura.

Visando a definição de uma sistemática a ser seguida no desenvolvimento de ambientes virtuais, Green, Halliday (1996) propõem uma estratégia que considera tanto a parte de definição de objetos em cena, quanto as interações e comportamentos destes objetos, composta por três fases:

1ª) Projeto de objetos de cena: com a definição da geometria e comportamento dos objetos no ambiente virtual;

2ª) Composição de Cenas: com a definição da localização dos objetos na cena do ambiente virtual, trabalhando com várias visões do ambiente e definindo as interações entre objetos;

3ª) Transições entre cenas: definição das mudanças de ambientes virtuais, dinamicamente.

De acordo com Costa (2000), a sistemática acima não está focada em requisitos específicos da área da aplicação, que é muito importante ao desenvolvimento de sistemas de RV de imersão subjetiva (Johnson, 1998). Uma outra abordagem, composta por quatro etapas modulares, é apresentada por Kirner, Martins (1998) e ilustrada pela Figura 4.1. Tal abordagem se aproxima do modelo em cascata destacado em Pressman (1995) e possibilita maior integração dos usuários com o desenvolvimento do produto.
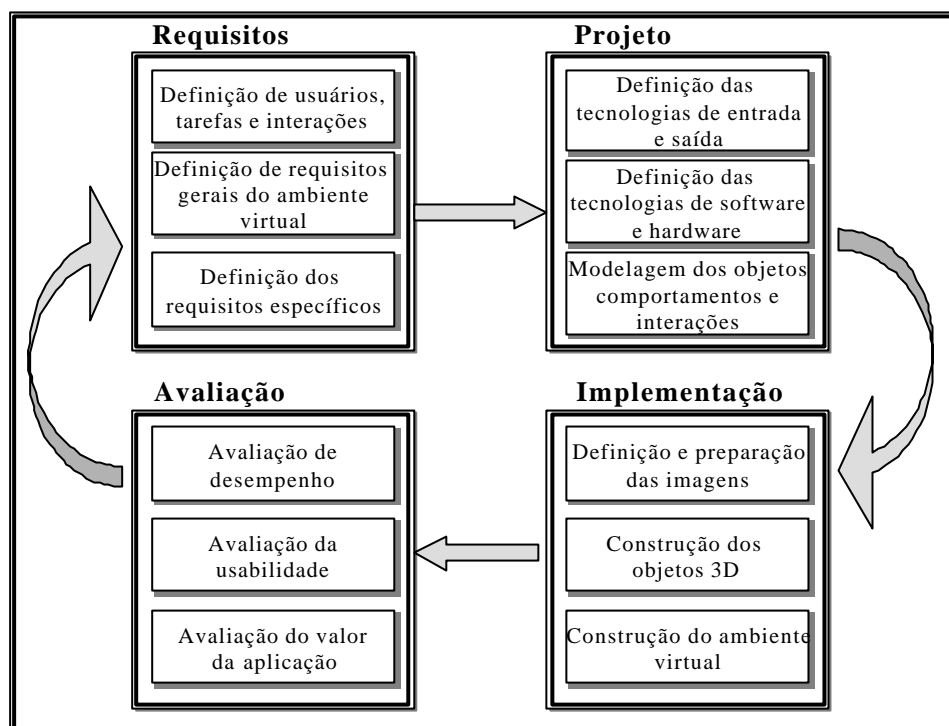
Fig. 4.1: Processo de Desenvolvimento de Ambientes Virtuais

Observa-se que esta proposta não contempla soluções relativas à concepção de um ambiente virtual que seja editor de outros ambientes virtuais (uma solução relativa a "meta-ambiente 3D"), capaz de permitir o desenvolvimento dinâmico de ambientes virtuais. Logo, são necessárias alterações no ciclo de vida proposto na Figura 4.1, de forma a adequá-lo às proposições desta forma de solução.

Nestes meta-ambientes 3D, um dos recursos disponíveis é a capacidade de permitir a associação dinâmica entre objetos de cenário e destes com os objetos de controle, por meio de definições providas pelo usuário (que é o responsável pela criação dos ambientes a serem produzidos). A Figura 4.2 apresenta uma proposta para o ciclo de vida deste tipo de desenvolvimento, a partir de alterações efetuadas no ciclo de vida representado pela Figura 4.1.
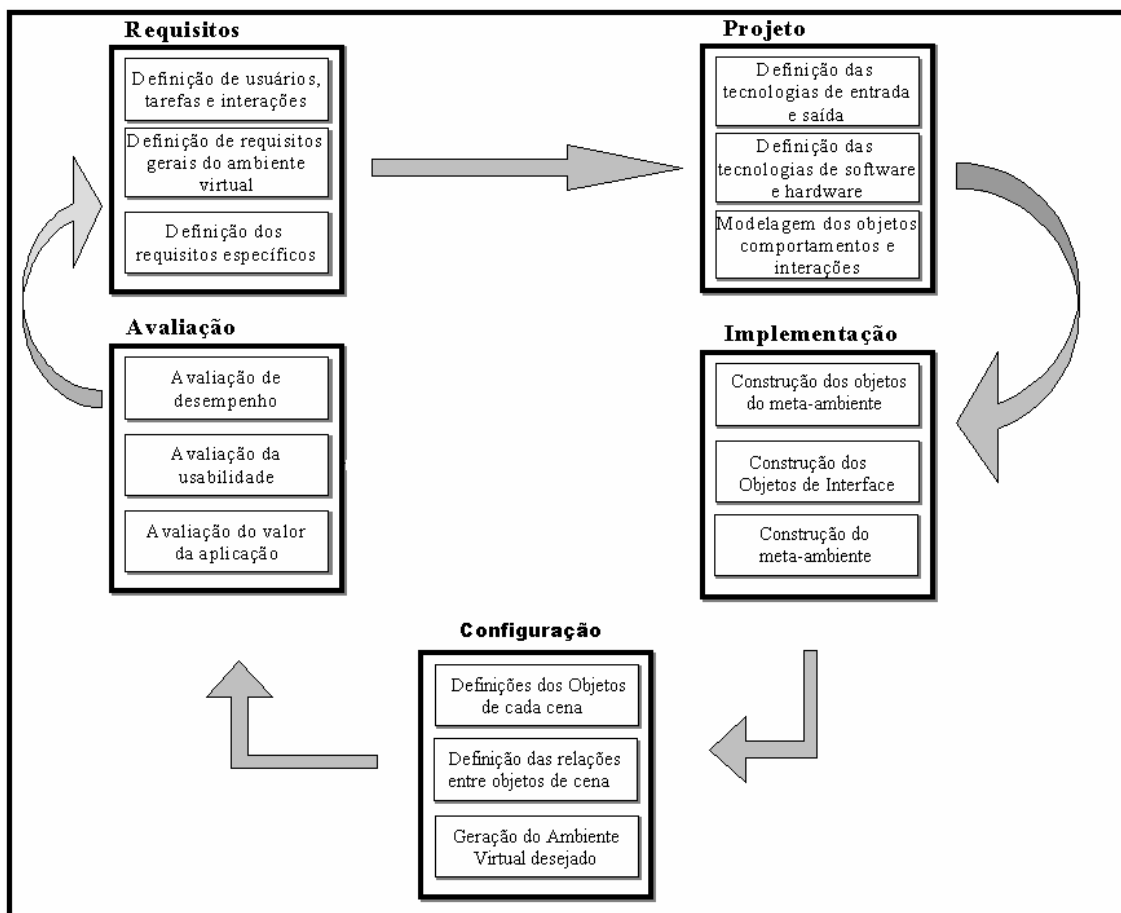
Fig. 4.2 - Alterações no Ciclo de Vida de Desenvolvimento.

As alterações estão relacionadas com diversas etapas do ciclo. Do ponto de vista da análise, visando facilitar a definição deste tipo de sistema, uma boa estratégia é, por meio da utilização da análise orientada a objetos, prover condições de melhoria da interação com o usuário final para permitir a definição dos conjuntos de atributos e operações (pertencentes ao seu domínio de conhecimento) de objetos do domínio da solução.

É necessária uma boa seleção de atributos, operações e classificações (pública, privada ou protegida) relativas a tais objetos, de forma a dar extensibilidade à definição do sistema e a atender requisitos. Aspectos relativos aos objetos necessitam ser amplamente discutidos com os usuários, que detêm importantes informações relacionadas com os mesmos e com as possibilidades de interações.

A concepção de meta-ambientes 3D exige uma acurada análise de riscos, já que o desenvolvimento de ambientes voltados a uma solução específica, relacionados com a

definição e desenvolvimento de ambientes virtuais para um fim específico, é mais previsível.

Na fase de projeto, do ponto de vista de modelagem, um bom equilíbrio entre a funcionalidade e a possibilidade de reutilização é decisivo na escolha das classes que comporão tais ambientes. Uma boa definição dos tipos de classes e das atribuições dos objetos, do processo de interação com o usuário e das relações entre objetos, favorece a modelagem geométrica e comportamental. É importante selecionar, por funcionalidade, as classes e definir aspectos da interação entre objetos e destes com os usuários.

Durante estas definições, podem surgir dúvidas quanto ao agrupamento dos objetos, já que as múltiplas interações tendem a tornar tais agrupamentos menos identificáveis. A evolução do ciclo de vida e a construção de protótipos tornam tais definições mais claras, com identificação das classes necessárias.

Para atingir tais objetivos, uma boa estratégia é definir objetos simples (instâncias de classes), relacionados a um caso de uso específico e estender a solução, visando satisfazer outros pontos do projeto do sistema, similar à estratificar para visualizar o sistema como um todo. Na estratificação, a interação com usuários permite definir, construir e avaliar rapidamente um dado protótipo. Como resultado, há uma melhor definição das classes, o que contribui para a coesão, flexibilidade do sistema e reutilização.

A partir da proposta de estratificação, uma saída é desenvolver algum(s) subproduto(s) como protótipo(s), validar o(s) mesmo(s) e avaliar os desafios para desenvolver novos subprodutos. A definição das classes, nesta estratégia, é *bottom-up*, de forma que a partir de objetos definidos no(s) subproduto(s) pode-se observar necessidades do sistema que se pretende desenvolver.

Os subprodutos apontam objetos necessários, relacionamentos entre objetos e destes com os usuários, além de permitir definições acerca de visibilidade de atributos e operações. No caso de atributos, elementos como o tipo (estático, constante ou normal), tipo de dados e domínio de valores podem ser melhor visualizados, uma vez que o usuário participa novamente do ciclo de vida através da avaliação do protótipo.

Uma das vantagens deste processo é relativa a minimizar as expectativas do usuário e do analista, pela distinção de limitações (como por exemplo, relativas a número de elementos, formas de visualização dos ambientes virtuais, formas de interação, tipos de

objetos de interface etc) e fronteira do sistema, provendo condições para a evolução na forma de projeto.

Na concepção de meta-ambientes de Realidade Virtual, uma avaliação de tecnologias de entrada e saída fundamenta-se na verificação de tecnologias acessíveis para potenciais usuários. Neste aspecto, motivos como a ampla utilização, o barateamento de equipamentos e vasta gama de periféricos disponíveis e flexibilidade tornam o computador pessoal uma boa escolha, ainda que pesem limitações de velocidade e extensibilidade a periféricos. De fato, a possibilidade de atingir um número cada vez maior de usuários deve ser um dos elementos da análise.

No caso de sistemas suportados pela Web, a grande diversidade de tecnologias esconde um grande número de limitações. Assim, uma boa definição do software, através de uma análise de outras soluções e das tecnologias comumente empregadas por diversos pesquisadores pode ser utilizado para a definição do(s) protótipo(s). Entretanto, estas tecnologias podem, em definições extensíveis, tornarem-se inviáveis.

Na fase de implementação, para permitir que os objetos sejam definidos em tempo de criação de um novo ambiente virtual, define-se uma biblioteca básica de objetos, a partir dos requisitos, que são modelados do ponto de vista geométrico e comportamental (na fase de projeto). São gerados, ainda, os objetos de interface, os objetos que figurarão no meta-ambiente e a configuração deste de forma a capacitar usuários desenvolverem um novo ambientes virtual.

O meta-ambiente e os ambientes virtuais gerados são avaliados, em termos de desempenho, usabilidade e do valor da aplicação, gerando uma apurada análise de riscos. Observa-se a forte participação dos usuários finais (professores, neste caso particular) no processo de desenvolvimento. Como resultados desta fase, novos objetos podem ser sugeridos e acrescentados, depurando a análise de requisitos e o modelo de análise.

Em novas voltas do ciclo, não haverá grandes alterações no projeto de classes, capaz de suportar novas modelagens de objetos, sem a necessidade de alterações nas tecnologias de software e hardware (que já foram validadas). A implementação gera estes novos objetos, que são reavaliados em conjunto e separadamente em nova fase de configuração.

A proposta acima reflete um ciclo de vida dinâmico capaz de permitir a manutenção do sistema através de seu aprimoramento e de uma análise de riscos constante. É

importante destacar que o sistema pode ter novas soluções de implementação, dependendo da necessidade e dos resultados dos projetos. Mas, ao mesmo tempo, definições da fase de análise e requisitos não necessitam ser refeitas.

No desenvolvimento do sistema proposto, serão cumpridas as diversas etapas apresentadas na Figura 4.2. Resultam da fase de implementação, protótipos que, durante o desenvolvimento do sistema podem vir a ser reutilizados ou descartados, visando melhorias de qualidade e redução de manutenção.

As seções seguintes detalham a etapa de Requisitos e de Projeto. No capítulo cinco são apresentados protótipos de testes que demonstram a viabilidade do projeto e no capítulo seis, as avaliações de trais protótipos por clientes do sistema, relacionada com a Fase de Avaliação.

4.1.4. Visão Geral do Sistema

Booch et al. (2000) sugerem que um sistema pode ser decomposto em uma coleção de subsistemas para decomposição da complexidade em partes quase independentes. Tais subsistemas em um nível de abstração mais baixo podem constituir, de fato, um novo sistema.

O software proposto, denominado Laboratório Virtual de Física (LVF), é constituído de dois subsistemas e de um conjunto de kits de experimentos, conforme representado na Figura 4.3

São componentes do sistema:

(a) Subsistema de Criação de Experimentos: que permite ao usuário, tipicamente um professor, acessar o sistema para criar experimentos virtuais de Física compostos por ambientes virtuais em que figuram duas áreas: cenários virtuais e painéis de controle, utilizando-se de conjuntos de objetos disponíveis no ambiente em uso (que é um meta-ambiente). Este usuário é denominado **autor**;

(b) Subsistema de Atualização de Bibliotecas de Objetos: que permite a criação, manutenção e ampliação da bibliotecas de objetos disponibilizados, que são utilizados na composição dos ambientes virtuais (relativos aos experimentos de Física). Tais modificações devem prover condições de melhoria dos elementos disponíveis nos kits

de experimentos. O usuário deste sistema é geralmente um conhecedor de linguagens de programação e computação gráfica, sendo denominado **desenvolvedor**
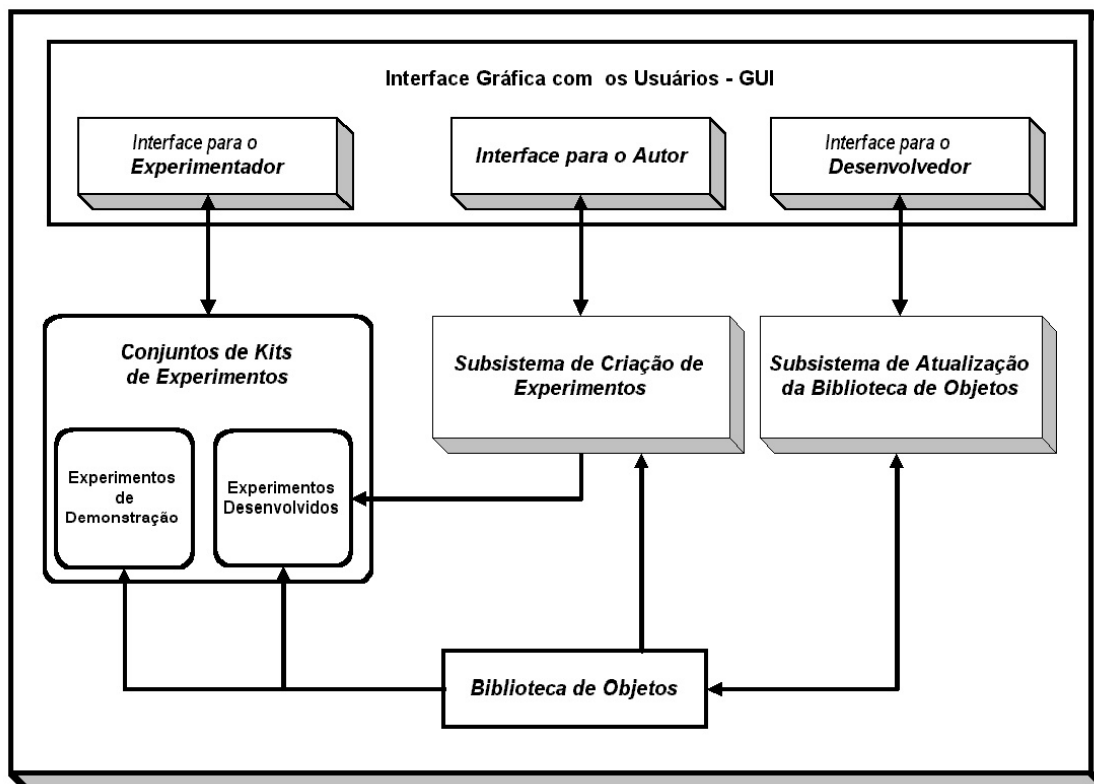
Fig. 4.3 - Visão Geral dos Componentes do Software LVF

(c) Conjunto de kits de Experimentos: relativo à biblioteca que contém os diversos experimentos disponíveis (na forma de ambientes virtuais), que podem ser de demonstração ou desenvolvidos pelos autores. Permite que um aluno (ou mesmo um professor) realize tais experimentos (virtuais) através da Internet, de forma a analisar e estudar aspectos relativos a um dado fenômeno físico. Este usuário é denominado **experimentador**.

Os diversos componentes do sistema estão descritos detalhadamente nas seções seguintes.

4.1.4.1. Subsistema de Criação de Experimentos

O Subsistema de Criação de Experimentos possibilita a criação de ambientes virtuais vinculados a experimentos, por meio de três áreas de trabalho distintas:

- **Cenário Virtual**: contém elementos 3D dispostos em um cenário, que caracterizam o experimento virtual, propriamente dito, a ser definido pelo autor. Tais elementos são inseridos pelo autor no cenário a partir da seleção na biblioteca de objetos de cenários e posterior definição de propriedades, por meio de objetos da interface gráfica;
- **Painel de Controle do Experimento**: contém elementos de interação (sensores e atuadores), definidos pelo autor, por meio de seleção destes componentes, usando objetos da interface gráfica. Tais objetos permitem ao experimentador o controle dos elementos que se encontram definidos no cenário virtual do experimento;
- **Ferramentas de Configuração:** contém *widgets* (menus, caixas de diálogo, listas etc) que possibilitam ao autor a edição e configuração dos elementos do cenário virtual e do painel de controle que comporão o ambiente virtual do experimento desejado.

O cenário virtual é composto dos seguintes tipos de objetos:

- **objetos de adorno**: usados para a composição do cenário, não participam ativamente dos experimentos e não exigem definições relativas a variáveis de experimentos (grandezas físicas). Neste grupo estão objetos como: fundo de tela, cadeiras, portas, quadros de adorno e outros;
- **objetos de experimento**: usados para compor os experimentos, exigem a definição de atributos relativos a características físicas e geométricas, de forma a definirem o comportamento adequado ao objeto quando inserido em um dado experimento. Alguns objetos deste grupo permitem a definição de uma escala de valores. Tais objetos são capazes de enviar e receber informações a elementos do painel de controle;
- **menus**, **sensores** e **atuadores 3D**: usados para trocas de informações com os objetos do cenário virtual, estando imersos no cenário. Provêm o envio e

recebimento de dados relativos aos objetos de experimentos ou ativam a carga de novas páginas web.

O painel de controle, é definido pelo autor e estará disponível na experimentação, se compõe dos seguintes objetos:

- **Sensores**: responsáveis pelo recebimento de informações (dados) dos objetos de experimento. Há sensores de velocidade, posição e outras propriedades físicas, adequadamente definidos para cada tipo de elemento de experimento, que via de regra, apresentam resultados numéricos.
- **Atuadores**: responsáveis pelo envio de ordens para objetos de experimento, são utéis no controle dos experimentos.

Na composição do painel de controle, o autor escolhe posição, aparência e escala de valores que um dado sensor ou atuador possuirá, de forma a personalizar tais componentes.

Durante a definição do cenário e do painel de controle devem ser estabelecidas as relações entre os diferentes objetos de experimento e destes com menus3D, sensores e atuadores, por meio das ferramentas de configuração, que figuram na interface para a criação de experimentos. Caso seja realizada uma tentativa incorreta de associação, o sistema informa ao autor e solicita a correção. Ao completar a definição de um experimento, o autor poderá salvá-lo, tornando-o disponível para acesso via Web (para posterior utilização pelo experimentador).

4.1.4.2. Subsistema de Atualização da Biblioteca de Objetos

O Subsistema de Atualização de Objetos de Experimentos permite a alteração, inserção e atualização dos elementos que compõem os cenários virtuais e os painéis de controle. A Figura 4.4 apresenta, esquematicamente, os componentes da Biblioteca de Objetos, que são passíveis de atualização.
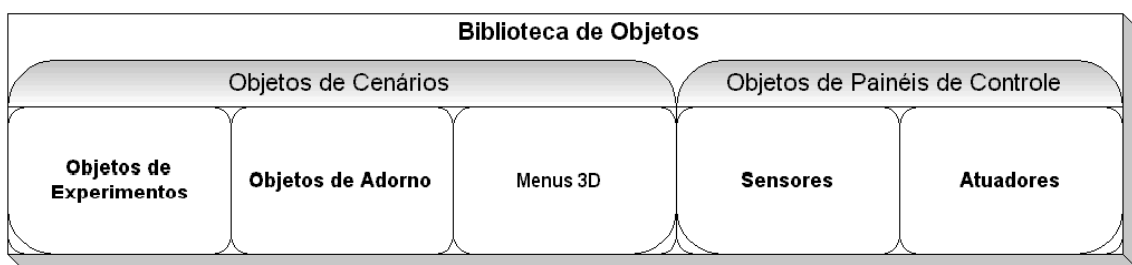
Fig. 4.4. Componentes da Biblioteca de Objetos

São características deste subsistema:

- capacidade de acesso a diretórios e arquivos relacionados com os diversos objetos da Biblioteca, através de uma área para visualização do diretório corrente e seleção do arquivo que contém as informações relativas á definição dos objetos de adorno, objetos de experimento, menus3D, sensores e atuadores;
- possuir um editor de textos, capaz de suportar a exibição dos códigos-fonte dos objetos de adorno, experimento e menus, que é usado para entrada do texto relativo a novos objetos e modificação dos textos relativos a objetos correntes ;
- incorporar um analisador sintático que verifica as modificações dos códigos, a partir de comandos dados pelo desenvolvedor, sendo capaz de evitar erros de sintaxe que possam inviabilizar a alteração ou inserção de código;
- uma interface que suporta o manuseio de arquivos, a edição de códigos e a atuação do analisador através de uma área para visualização dos requisitos de objetos, que apresenta dados relativos a estes objetos, de forma a evitar definições incorretas ou duplicadas.

É permitida a alteração de códigos de objetos e a inserção de novos códigos, que são validados pelo subsistema. Não havendo erros, tais objetos são inseridos no diretório correspondente.

4.1.4.3. Conjunto de Kits de Experimentos

Este conjunto, utilizado pelo experimentador, é composto de ambientes virtuais (de demonstração ou desenvolvidos por autores) que permitem:

- a escolha do experimento que será simulado, dentre um conjunto de experimentos previamente definido;
- a interação com o experimento escolhido, através dos elementos do cenário virtual (interações diretas) ou dos elementos do painel de controle (interações indiretas).

Cada experimento tem a configuração dada na sua definição, apresentando ambos: um cenário virtual e um painel de controle. As propriedades dos objetos, relações entre eles e com sensores e atuadores não podem ser modificadas na fase de experimentação. No entanto, o experimentador poderá variar valores associados aos elementos do experimento, caso tal recurso tenha sido previsto na definição.

Para utilização dos experimentos virtuais, o experimentador deve:

- escolher o tópico de interesse, ao acessar o software, através da escolha do kit de experimento desejado, que é composto dos ambientes virtuais desenvolvidos e relacionados àquele tópico;
- escolher o experimento que deseja desenvolver dentre os disponíveis no kit;
- interagir com o ambiente virtual de forma direta, através dos elementos do cenário virtual que lhe são apresentados, podendo alterar a visualização, navegar entre os objetos, verificar e atualizar propriedades físicas dos mesmos;
- interagir com elementos do painel de controle (interações indiretas) de forma a visualizar resultados, promover alterações de valores relativos a variáveis de experimentos e disparar a inicialização de processos relativos aos experimentos e elementos do cenário 3D.

4.1.5. Requisitos: Identificação de Casos de Uso do Sistema

Durante a fase de análise há a necessidade de geração de modelos do sistema para permitir a avaliação e o projeto (Furlan, 1998). O modelo de análise especifica objetos lógicos que serão incluídos no sistema, destacando seus grupos e relações. Associado a ele, há o modelo de requisitos que destaca a funcionalidade do sistema a partir do uso de atores e de casos de uso. Assim, para modelar o sistema serão levantados os aspectos funcionais desejados e, posteriormente, os aspectos relativos às limitações e comportamentos.

4.1.5.1. Casos de Uso para o Autor

De forma a definir as funcionalidades do sistema, é necessária a definição dos casos de uso e dos atores. A Figura 4.5 apresenta os casos de uso relacionados com o autor, geralmente, um professor que deseja elaborar experimentos virtuais de Física, com a utilização do software Laboratório Virtual de Física - *LVF*.
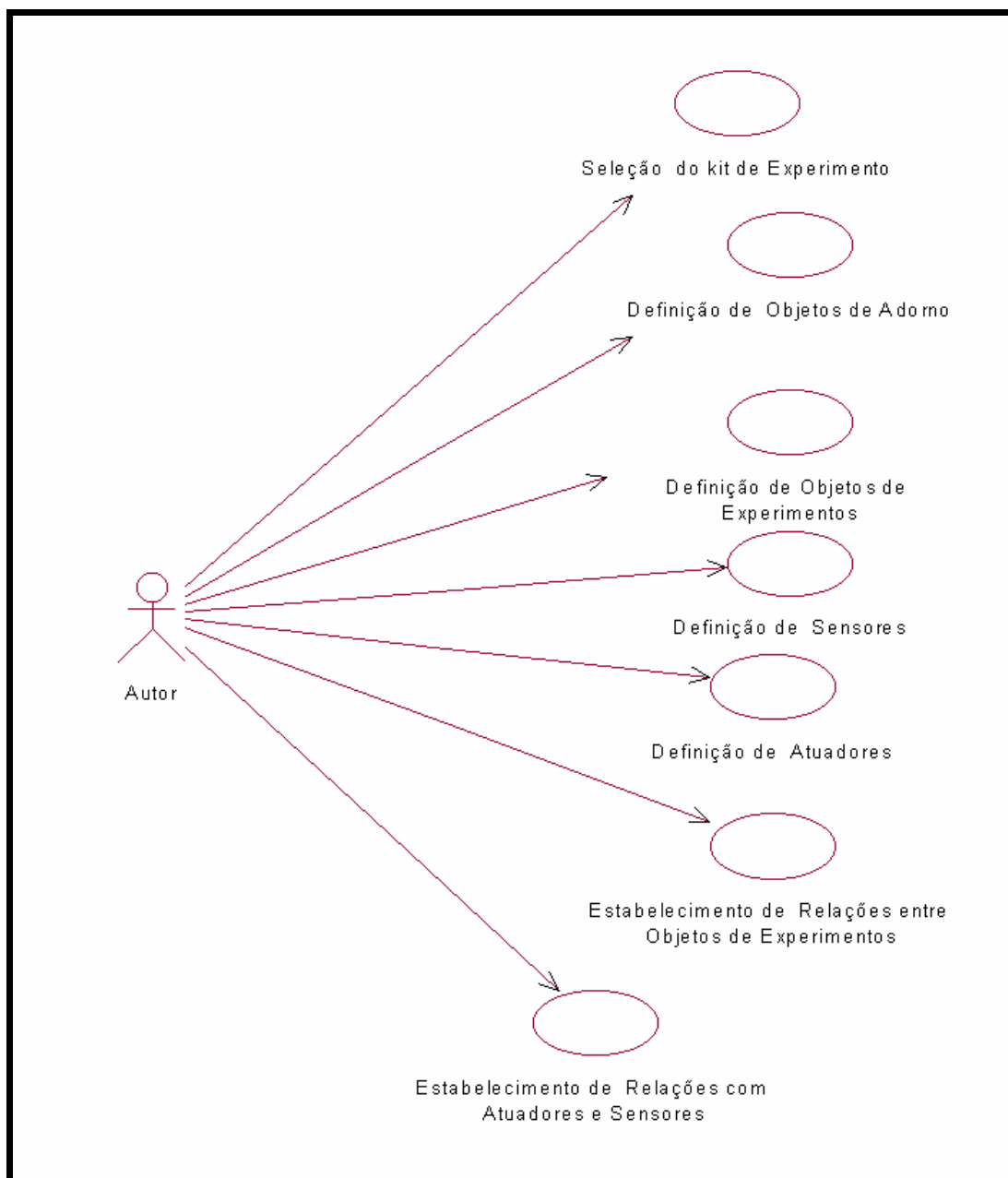
Fig. 4.5 - Casos de uso do software LVF para o autor

Descrição do caso de uso "Seleção do kit de Experimento":
▪ *Fluxo de eventos principal*: O caso de uso começa pela seleção de uma dado kit de experimento, através da interface inicial do sistema. O sistema solicita o nome do usuário e uma senha em uma caixa de diálogo. O autor confirma a entrada pressionando a tecla Enter. O sistema carrega uma página Web composta por um ambiente virtual e

um painel de controle por definir, além das Ferramentas de Configuração (um conjunto de *widgets*) para definição dos objetos e das relações.

- *Fluxo excepcional de eventos:* Se o autor introduzir um nome de usuário ou senha inválidos, o sistema solicita novamente estes dados. Se isso acontecer por três vezes consecutivas, o sistema cancela a transação.

Descrição do caso de uso "Definição de Objeto de Adorno":

- *Fluxo de eventos principal:* O caso de uso começa com a seleção de um objeto de adorno associado a uma *widget*. O sistema solicita atributos do objeto de adorno relativos a seu posicionamento e sua aparência. O autor define tais atributos e pressiona a tecla Enter. O objeto é inserido no ambiente virtual em desenvolvimento.
- *Fluxo excepcional de eventos:* Se o autor definir valores absurdos para atributos do objeto ou omitir valores necessários, uma mensagem de erro é apresentada, solicitando a correção de dados e o objeto não é inserido no ambiente virtual.

Descrição do caso de uso "Definição de Objeto de Experimento":

- *Fluxo de eventos principal:* O caso de uso inicia-se com a seleção de um objeto de experimento associado a uma *widget*. O sistema solicita atributos deste objeto relativos à posicionamento, aparência, propriedades físicas e relações com outros objetos de experimento, que já tenham sido definidos e inseridos no cenário virtual. Para definir as relações, uma lista apresenta os possíveis objetos de experimento que podem ser associados àquele que está em definição. O autor define valores e relações e pressiona Enter. O sistema insere o objeto no ambiente virtual em desenvolvimento.
- *Fluxo excepcional de eventos:* Se o autor definir valores absurdos para atributos físicos do objeto ou omitir valores necessários, uma mensagem de erro é apresentada, solicitando a correção de dados e o objeto não é inserido no ambiente virtual.

Descrição do caso de uso "Definição de Sensores":

- *Fluxo principal de eventos:* O caso de uso inicia-se quando o autor seleciona um sensor através de uma *widget*. O sistema solicita a definição do local, dentro do painel de controle, onde o sensor será localizado e o objeto com quem o sensor trocará mensagens. Como os objetos podem enviar mensagens diversas, para cada tipo de objeto de experimento selecionado, uma lista de valores apresenta quais atributos

podem ser capturados pelo sensor e a escala em que estes devem figurar. Definido este relacionamento, o autor pressiona Enter. Se o sensor e a relação definida forem válidos, o sistema insere-o no painel de controle.

- *Fluxo excepcional de eventos:* Se o autor tentar uma associação de sensor com objeto de experimento não válida, é apresentada uma mensagem de erro e solicitada a correção.
- *Fluxo excepcional de eventos:* Se a escala definida pelo autor não for compatível com a informação que é enviada pelo objeto de experimento associado, uma mensagem de erro é apresentada, impedindo a inserção do sensor no painel de controle até a devida correção.

Descrição do caso de uso "Definição de atuadores":
▪ *Fluxo de eventos principal:* O autor seleciona, por meio de uma *widget*, um atuador. O sistema lista os objetos de experimento que permitem atuações e solicita que o autor defina a relação atuador-objeto. Além disto, solicita atributos relativos ao nome, posição e aparência do atuador. Se os valores e as associações forem válidos, o sistema insere o atuador no painel de controle. ▪ *Fluxo excepcional de eventos:* Se o autor definir uma posição para o atuador no painel de controle, que não permita uma boa visualização do mesmo, uma mensagem de erro é apresentada e o atuador não é inserido no painel de controle.

Descrição do caso de uso "Estabelecimento de Relações entre Objetos de Experimentos":
▪ *Fluxo de eventos principal:* O caso de uso começa quando o autor seleciona a opção relativa à nova associação entre objetos de experimento. O sistema apresenta uma lista com todos os objetos de experimentos e uma segunda lista vazia. Quando o autor seleciona um dado elemento na primeira lista, são apresentados na segunda lista todos os objetos com as quais este objeto pode se relacionar. Selecionado o elemento na segunda lista, uma solicitação de associação entre objetos é apresentada. O autor deve definir a forma de relação através desta solicitação. Se a relação for válida, o ambiente é alterado de forma a atender à esta definição. ▪ *Fluxo excepcional de eventos:* Se o autor tentar definir uma forma de relação não válida entre dois objetos, o sistema não processa a solicitação e envia uma mensagem de erro.

Descrição do caso de uso "Estabelecimento de Relações com Atuadores e Sensores":

- *Fluxo de eventos principal:* O caso de uso começa quando o autor seleciona a opção de nova associação com sensor/atuador. O sistema lista os sensores e atuadores em uma lista e apresenta uma segunda lista vazia. Com a seleção de um dos sensores ou atuadores nesta lista, a segunda lista apresenta os objetos com as quais este elemento pode ser associado, considerando os atributos dos objetos de experimento e o tipo de sensor/atuador selecionado. O autor deve selecionar um ou mais elementos da segunda lista e pressionar Enter. O sistema cria rotas de trocas de informações entre o sensor ou atuador e o(s) objeto(s) escolhidos para envio e recebimento de informações na forma de mensagens.
- *Fluxo excepcional de eventos:* Se o autor tentar definir relações do objeto com elemento do painel de controle com a qual não seja possível a associação, uma mensagem de erro é apresentada, solicitando verificação e uso de arquivos de ajuda do sistema.

4.1.5.2. Casos de Uso para o Experimentador

A Figura 4.6 apresenta os casos de uso relacionados com o Experimentador.
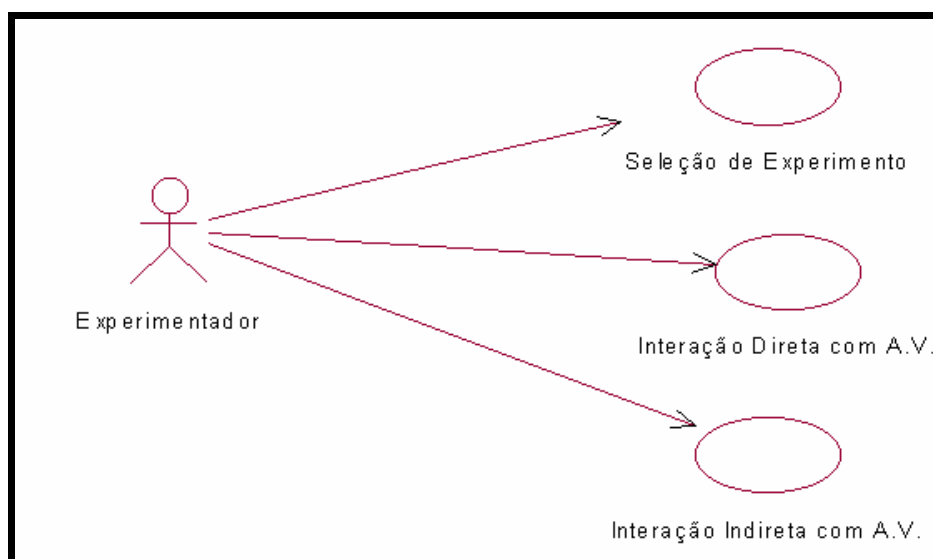
Fig. 4.6 - Casos de Uso para o Experimentador

Descrição do caso de uso "Seleção de Experimento":
▪ *Fluxo de eventos principal:* O caso de uso começa quando o experimentador acessa o sistema e, por interação direta com elementos disponíveis no cenário virtual, escolhe um kit de experimentos. Após tal escolha, o sistema atualiza o cenário e apresenta novas opções de escolha relativas ao kit selecionado. O experimentador, novamente por interação direta com objetos do cenário virtual escolhe o experimento desejado. O Ambiente Virtual relativo a tal experimento é apresentado ao Experimentador através da mesma janela do navegador.

Descrição do caso de uso "Interação Direta com A.V.":
▪ *Fluxo de eventos principal:* O caso de uso começa quando o Experimentador resolve disparar um processo físico relativo ao experimento que está visualizando. Através do toque em elementos do cenário, a alteração do tipo de cursor sinaliza a possibilidade de interação direta com este elemento. O experimentador clica ou arrasta o elemento (conforme for permitido pelo mesmo) e atualiza um atributo como desejar ou dispara uma operação.

Descrição do caso de uso "Interação Indireta com A.V.":
▪ *Fluxo de eventos principal:* O caso de uso está relacionado com a utilização do painel de controle (que é parte integrante do A.V. e figura em todo experimento). O Experimentador interage com elementos do painel de controle e promove a alteração de atributos de um dado objeto do cenário virtual ou a execução de um dado processo.

4.1.5.3. Casos de Uso para o Desenvolvedor

A Figura 4.7 apresenta os casos de uso relacionados com o Desenvolvedor.

Descrição do caso de uso "Consultar Biblioteca de Objetos do Sistema:"
▪ *Fluxo de eventos principal:* O caso de uso começa quando o Desenvolvedor através da interface de desenvolvimento seleciona uma dada classe para visualizar suas características. O sistema apresenta em uma área específica da interface a classe selecionada, destacando atributos e métodos pertinentes a esta. A partir da visualização de tais elementos, o Desenvolvedor pode compor o código relativo a um novo objeto desta classe.

- *Fluxo de eventos excepcional:* Se o Desenvolvedor tentar modificar elementos da classe que está sendo apresentada, o sistema informa que tal operação não é possível.
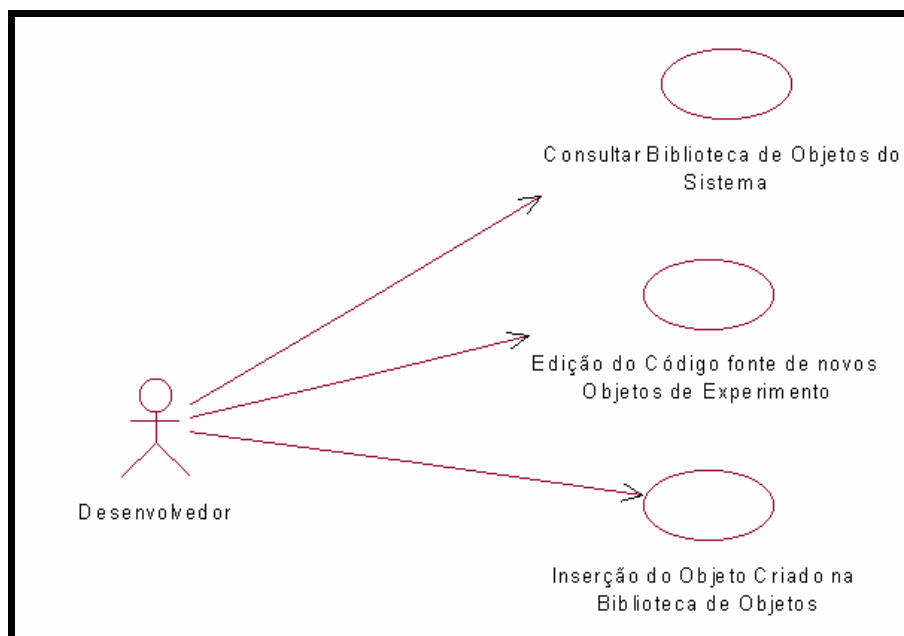
Fig. 4.7 - Casos de Uso para o Desenvolvedor

Descrição do caso de uso "Edição do Código Fonte de novos Objetos de Experimentos:"

- *Fluxo de eventos principal:* O caso de uso começa pela edição de um código para um novo objeto, seguindo as características da classe a qual este pertencerá. O Desenvolvedor pode importar um arquivo texto através dos objetos da interface ou editar todo o arquivo referente ao código dentro da área de texto.
- *Fluxo de eventos excepcional:* Se o Desenvolvedor promover erros de sintaxe ao digitar o texto, o sistema sinaliza erro de sintaxe através de *widgets* específicas. Neste caso, não estará habilitada a função de salvar arquivo de objeto.

Descrição do caso de uso "Inserção do Objeto Criado na Biblioteca de Objetos":

- *Fluxo de eventos principal:* O caso de uso está relacionado ao fim da edição do código fonte do objeto que será inserido (que deve estar livre de erros). O Desenvolvedor seleciona a opção de salvar o arquivo. O sistema verifica erros que porventura existam e solicita o nome do objeto, para inseri-lo na Biblioteca de Objetos.

- *Fluxo de eventos excepcional:* Se o arquivo ainda contiver erros, uma mensagem é apresentada e o sistema não salva o mesmo.
- *Fluxo de eventos excepcional:* Se o nome digitado pelo Desenvolvedor já for utilizado por outro objeto, o sistema informa que já existe tal objeto e não permite que sejam salvas as informações. Neste caso, será necessário a modificação do nome de forma a poder ser salvo o arquivo referente ao novo objeto.

4.1.6. Arquitetura Básica

A Figura 4.6 apresenta a arquitetura básica do sistema. Na seqüência são descritos os diferentes componentes do sistema e as interfaces gráficas com o(s) usuários(s), que desempenham importante papel na extensibilidade do mesmo.
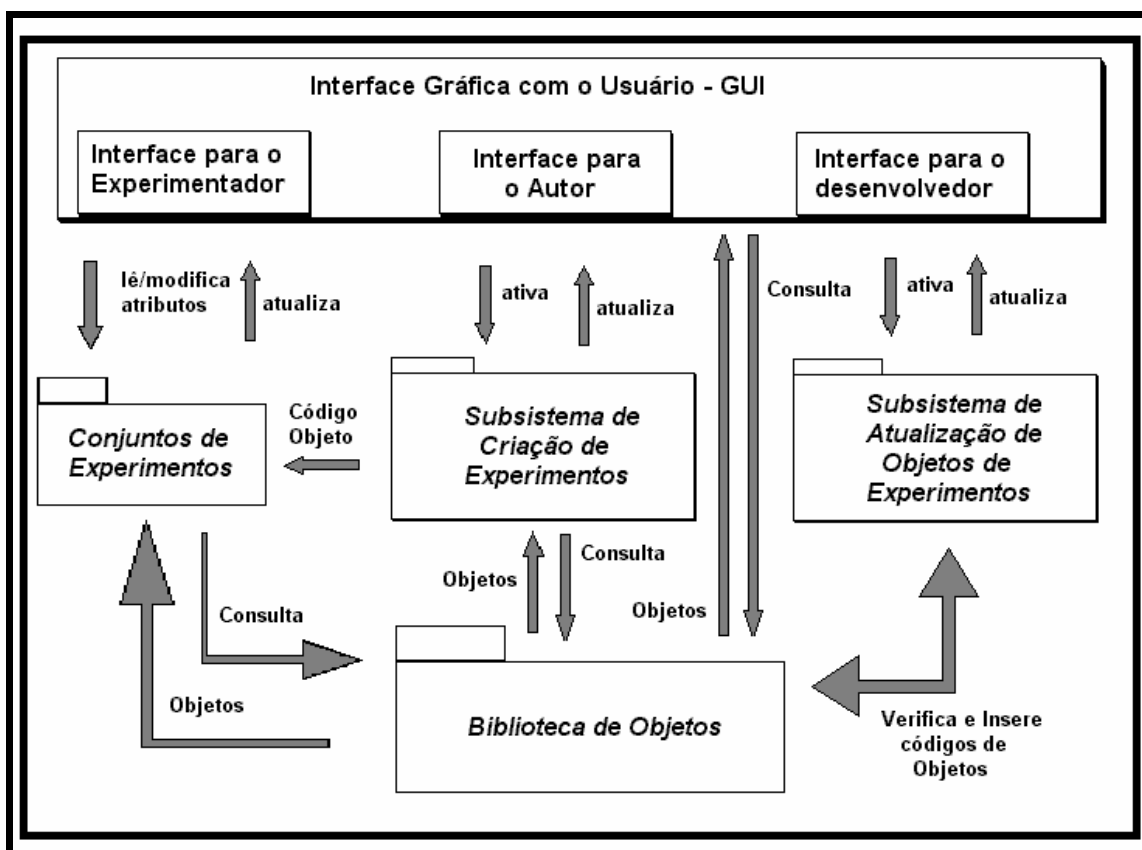
Fig. 4.6 - Arquitetura Básica do Sistema

4.1.6.1. Interface Gráfica com o Usuário - GUI

A Interface Gráfica com o Usuário - GUI representa a interface (do meta-ambiente e

dos ambientes desenvolvidos) e deve suportar as tarefas de criação, desenvolvimento e experimentação, sendo compatível com a Web.

É tarefa da GUI gerenciar as páginas Web que são apresentadas ao usuário e coordenar as ações de ativação de outras páginas, subsistemas, aplicativos e de conjuntos de instruções. Inicialmente, a qualquer usuário do sistema é apresentado o cenário de um laboratório virtual (3D), que contém um conjunto de pontos de vista pré-definidos. Tal cenário é capaz de permitir a escolha entre criação de um novo experimento, a atualização da biblioteca de objetos ou a ativação de um dado experimento pré-definido. A forma de interação do usuário com elementos deste ambiente define que componentes serão ativados.

As características da interfaces ativadas, a partir desta interface inicial, variam com a intenção do usuário no acesso ao sistema. Caso o usuário opte por carregar um experimento disponível, a GUI através da Interface de Experimentação contém *widgets* capazes de prover a seleção do experimento desejado (baseado em frentes da Física disponíveis).

Quando o usuário opta por desenvolver um novo experimento, a GUI, através da Interface de Criação, aciona o Subsistema de Criação de Experimentos. Se o usuário optar por modificar os objetos, a GUI ativa o Subsistema de Atualização de Objetos de Experimentos que provê funções para a verificação e definição de novos objetos (modificando o pacote de Objetos de Cenários).

Dentre as características comuns das diferentes interfaces com os usuários, destacam-se que os objetos de interface - *widgets*, que possibilitam ações do usuário de forma natural dentro do cenário 3D apresentado, abstraindo seu comportamento dentro de um laboratório real com interações diretas, tais como: tocar um objeto, agarrá-lo e colocá-lo sobre uma mesa de experimento.

O autor, por interação direta com elementos do cenário e indireta com as *widgets* pode definir novos ambientes virtuais para seus experimentos, modificá-los e armazená-los, ou acessar demonstrações do sistema e a ajuda *on-line* (*help*).

Há situações em que são necessárias as definições de atributos dos objetos de cenário, como posição de um elemento, sua massa etc. Nesta hipótese, por interação indireta, o autor deve usar periféricos de entrada de dados (como teclado, mouse) para inserir tais valores (perdendo parcialmente sua imersão no sistema).Visando solucionar tais

interações, a GUI apresenta um painel que as provê.

A interface de experimentação, relacionada com o experimentador, se relaciona com o pacote de experimentos armazenados e pode ler e modificar atributos (se tal característica foi definida para o AV) de objetos existentes nos ambientes virtuais relativos a tais experimentos.

Ao final deste capítulo, uma descrição detalhada das interfaces será apresentada.

4.1.6.2. Conjuntos de Experimentos

O componente relativo aos Conjuntos de Experimentos organiza os experimentos criados e disponíveis em pastas e subpastas. A seleção da pasta e subpasta relativa à cada experimento é provida a cada usuário por elementos da interface, de forma que o acesso ao conteúdo de uma dada aula pelo experimentador está associado à escolha de um elemento do cenário que se relaciona com um dado kit de experimentos e em seguida com a aula desejada. Por exemplo, há uma pasta de experimentos relacionados com a cinemática que contém outra pasta de experimentos relativos a movimentos uniformes. Cada experimento está relacionado a um arquivo específico que contém o ambiente virtual equivalente.

A organização destas pastas e dos arquivos a elas relacionados é uma das funções do Subsistema de Criação de Experimentos. No acesso a este, são feitas seleções relativas às frentes da Física e por conseqüência o subsistema tem condições de controlar em que pasta será armazenado cada novo AV desenvolvido.

4.1.6.3. Subsistema de Criação de Experimentos

O Subsistema de Criação de Experimentos permite ao autor elaborar um experimento virtual como se estivesse dentro de um laboratório real. Do ponto de vista da interface, há uma janela onde pode ser visto o experimento que está sendo montado, o painel de controle e outra janela que contém comandos de inserção/remoção e definição de objetos.

Este sistema se relaciona com o pacote de Objetos dos Ambientes Virtuais, que contém as definições dos objetos, de forma a obter as informações dos objetos que o autor pode utilizar.

A Figura 4.7 representa a arquitetura do Subsistema de Criação de Experimentos, descritos por:
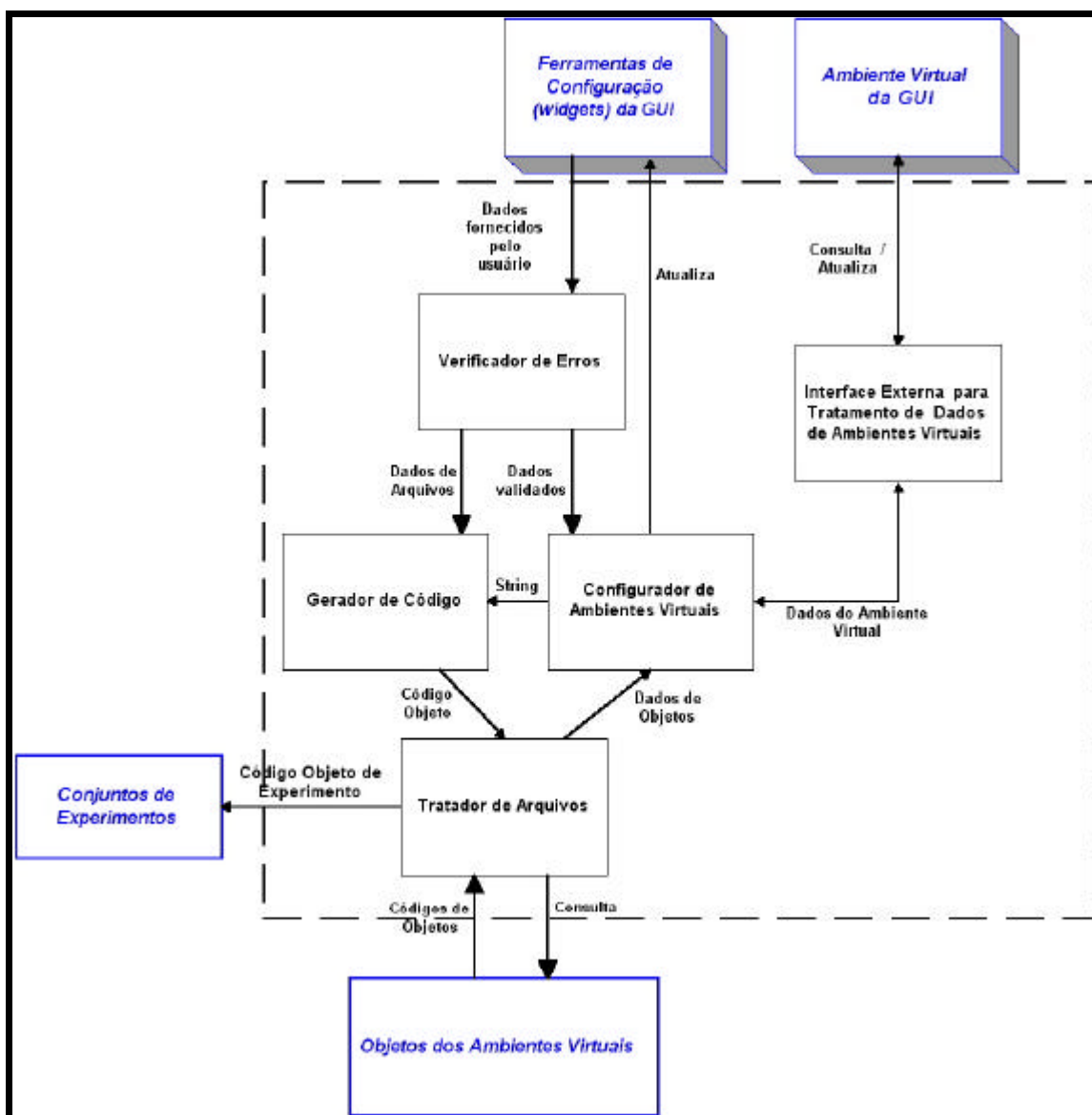
Fig. 4.7 - Arquitetura do Subsistema de Criação de Experimento

- Verificador de Erros: capaz de analisar os dados fornecidos pelo usuário e validá-los de forma a evitar entradas de dados incorretas que inviabilizariam o processo de definição do AV relativo ao experimento virtual;
- Configurador de Ambientes Virtuais: módulo de configuração da *string* temporária que será convertida em código de um experimento (código objeto). Este módulo recebe os dados de entrada do usuário (através dos objetos de interface e do AV corrente) e os dados dos arquivos que contêm os objetos,

verifica as associações definidas pelo usuário e promove a atualização do ambiente virtual e dos dados de entrada para a GUI. Cabe a este módulo o tratamento de problemas específicos de Computação Gráfica, tais como a associação de transformações gráficas;

- Gerador de Código: a partir das definições da *string* que contém as características do AV produzido, gera o código objeto correspondente a um dado experimento, consultando dados relativos ao experimento fornecidos pelo autor;
- Uma Interface Externa de troca de informações dos objetos da Interface de Criação (*widgets*) com os objetos do Cenário 3D que está sendo desenvolvido (Interface 2Dx3D). Esta interface é constituída por objetos capazes de interpretar os comandos do usuário e atualizar a visualização dos objetos do ambiente virtual. Tal interface ainda é capaz de receber dados enviados pelos objetos de cenário e informá-los ao autor. Assim, quando, por manipulação direta um dado objeto é arrastado pelo autor no cenário 3D, a informação de tal alteração é passada para o Configurador.

4.1.6.4. Subsistema de Atualização de Objetos de Experimentos

O Subsistema de Atualização de Objetos de Experimentos está relacionado diretamente com as classes e objetos que compõem o Cenário 3D e o Painel de Controle. Permite melhorar a biblioteca de objetos disponíveis para o autor, por inserção de novos objetos ou alteração de objetos já disponíveis.

Sua interface é mais complexa e exige usuários mais familiares a conceitos relativos à programação e orientação a objetos. Tal usuário, obedecendo os protocolos do sistema, tem condições de manutenção e ampliação de códigos de objetos, como contemplado no ciclo de vida apresentado na Figura 4.2. A Figura 4.8 apresenta os diversos componentes deste subsistema.

De forma a controlar tais alterações, desenvolvedores possuem senha específica para acesso ao sistema. Um arquivo de *log* permite o controle dos acessos efetuados por cada um e contém as informações do acesso. O Módulo de Validação do Usuário controla os usuários que têm acesso. Caso usuários não cadastrados tentem acessar o sistema, este módulo não permite o acesso ao módulo Editor de Textos.
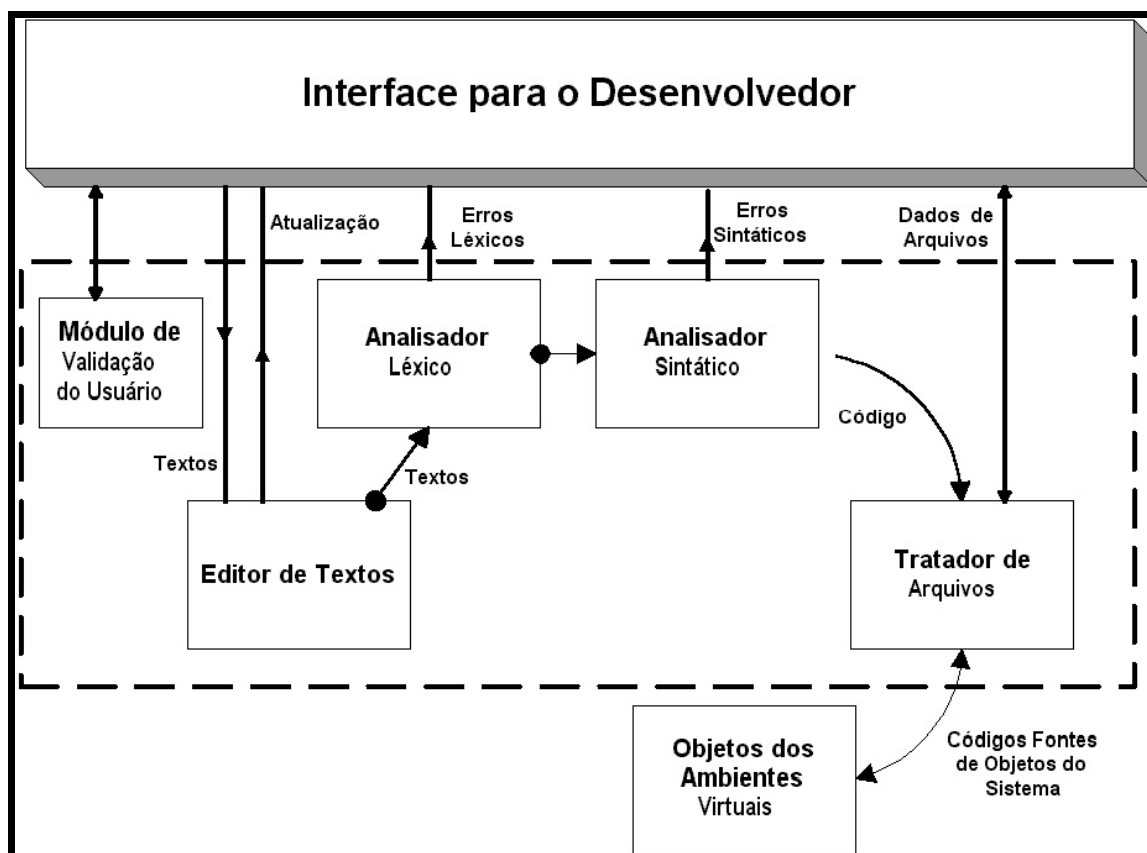
Fig. 4.8 - Arquitetura do Subsistema de Atualização de Objetos

O Editor de Textos é responsável pela aquisição dos textos fornecidos pelo usuário e pela atualização dos dados deste texto na GUI. Este módulo envia os arquivos de entrada ao Analisador Léxico, que promove a verificação ortográfica do texto que está sendo inserido ou alterado pelo usuário.

O Analisador Sintático é responsável pela validação sintática deste texto, analisando as diversas classes do sistema e o arquivo texto fornecido pelo desenvolvedor. O Código validado é enviado ao Tratador de Arquivos.

O Tratador de Arquivos permite o acesso aos arquivos fontes com dados relativos aos objetos já disponibilizados (implementados) e atualiza o Conjunto de Objetos dos Ambientes Virtuais com novos códigos, colocando-os em pastas adequadas, de acordo com entradas de dados do desenvolvedor.

4.1.6.5. Biblioteca de Objetos

A composição dos ambiente virtuais se dá com o uso dos Objetos dos Ambientes Virtuais (relativos a objetos de experimentos, objetos de adorno, menus3D, sensores e atuadores) disponibilizados pela Biblioteca de Objetos, que corresponde ao banco de objetos do sistema. A opção por um banco de objetos se deve ao fato de que um objeto pode ser tratado como uma caixa preta, que pode ser modificado e construído independentemente do sistema, desde que suas porções públicas não se modifiquem, permitindo a herança de atributos e operações (Furlan, 1998).

A biblioteca de objetos contém objetos que são usados na composição dos cenários dos experimentos, além da concepção de algumas interfaces, como a interface inicial do sistema, que é dada pela representação de um laboratório. Como a herança é uma das características deste banco, esta estratégia possibilita a criação de objetos a partir de objetos existentes, que transmitem suas características para um outro objeto, provendo condições de atualização de tal banco pelo desenvolvedor. Por outro lado, com a utilização de tais objetos, o autor pode definir as relações entre objetos de experimentos e destes com sensores e atuadores, através da GUI.

Tais relações permitem definir grupos de objetos associados e troca de mensagens entre os mesmos, de forma a se compor as propriedades físicas pertinentes a cada agrupamento ou elemento.

Para a concepção de experimentos virtuais será necessária a definição de posicionamento, tamanho, definição de propriedades físicas e relacionamento dos elementos do mundo que estarão disponibilizados na biblioteca. O sistema consulta a biblioteca de objetos do domínio, e a partir dela, escreve a versão final do código relacionada a um dado experimento definido.

4.1.7. Modelo de Objetos do Domínio

Para permitir a concepção do sistema, é necessária a definição das classes do domínio do mesmo, que são:

- Classe de Objetos de Adorno: Contém os diversos modelos geométricos de objetos utilizados no mobiliário dos laboratórios virtuais, tais como fundos de cenas, mesa, cadeiras, pranchetas, armários, portas etc.

- Classe de Objetos de Experimentos: contém os objetos que serão usados nas montagens experimentais, tais como carrinhos, molas, cestas. Nesta classe, há duas subclasses diferentes: objetos elásticos (para os quais será necessária a definição de uma constante elástica) e objetos duros.
- Classe de Sensores: É composta pelos objetos destinados à medição de variáveis dos experimentos físicos. São exemplos: velocímetros, dinamômetros, balanças de massa, dentre outros. Estes objetos necessitam trocar mensagens com objetos de experimento, de forma a obtenção atualizada de valores dos atributos dos mesmos.
- Classe de Atuadores: É composta por objetos que enviam mensagens a objetos de experimentos a partir de informações e interações com os usuários.
- Objetos de Interface 3D: É composta por menus 3D e por elementos de cenário utilizados para prover a ativação de experimentos por seleção do usuário. Nesta condição estão armários e gaveteiros que são capazes de prover a ativação de um dado experimento através da escolha do usuário.
- Classe de Objetos da Interface 2D: Contém os objetos relacionados com a interação com o usuário, que podem receber informações de entrada e enviar mensagens para outros objetos, ativando operações relacionados com a alteração de atributos dos mesmos relativos a um dado experimento. Serve ainda para coordenar a atualização das páginas que estão sendo exibidas. Um exemplo de objetos desta classe pode ser dado por menus de controle que permitem a um dado usuário modificar parâmetros dos objetos de cenário ou selecionar um dado tipo de experimento para ser desenvolvido.

O Anexo I apresenta o diagrama de classes do sistema modeladas de acordo com a UML.

4.2. Interfaces do Sistema

A interface do sistema será toda desenvolvida para a *World Wide Web* na Internet, definida por conjuntos de telas (páginas Web) com ambientes virtuais de imersão subjetiva e *widgets* de manipulação direta de elementos de cenários, provendo facilidades de criação ou reutilização de mundos virtuais que apresentam objetos inter-relacionados.

Para que o usuário possa desenvolver suas próprias páginas com as experiências desejadas, o sistema deverá possuir arquivos de auxílio, contendo dicas e informações que se apresentarão conforme a posição do cursor na página que está sendo utilizada pelo usuário. Como a interface estará fundamentada em páginas Web, o usuário estará experimentando navegar enquanto desenvolve seus propósitos.

Analisando as características comuns destas interfaces, as interfaces do sistema são constituídas por um Cenário 3D (contendo o experimento virtual já desenvolvido ou em desenvolvimento) que contém os objetos dos experimentos e com os quais o usuário pode interagir - interações diretas que se dão por manipulação dos objetos da cena (projeções perspectivas de um objeto na tela do computador).

Interações indiretas serão suportadas por janelas auxiliares que não compõem o cenário 3D apresentado, devendo estarem confinadas ao painel de controle ou a áreas externas ao ambiente virtual. Para prover as manipulações de dados dessas janela(s) auxiliar(es), serão usadas técnicas para posicionamento, com uso de seleção indireta através do mouse, técnicas para definição de textos com entrada direta (através do teclado alfanumérico), técnicas de quantificação através do uso de *scrollbar* e técnicas por apontamento com uso de menus.

Para suportar dúvidas dos usuários, arquivos de ajuda podem ser acessados ao longo da utilização e ativarão novas páginas Web de instrução. Os menus utilizados contêm itens habilitados ou desabilitados (de forma a evitar erros do usuário e visando usabilidade).

A Figura 4.9 apresenta um exemplo de interface inicial para o caso de uso de experimentação e desenvolvimento de novos experimentos, que é parte do protótipo do sistema. Composta pela simulação de um laboratório, com portas de vidro e mesas dispostas ao longo de uma sala apresentada no Cenário 3D equivalente, há mesas que contêm representações dos diversos kits de experimentos, onde o usuário opta por carregar ou elaborar um novo experimento.

Fig. 4.9 - Interface Inicial do Protótipo do Sistema

As seções seguintes destacam as variações da interface adequadas a cada caso de uso do sistema (apresentados anteriormente neste capítulo).

4.2.1. Interface de Criação

A interface será uma abstração - com representação 3D - de um laboratório constituído por armários qualificados por nomes equivalentes a tópicos da Física. São elementos do cenário: cadeiras, mesas, portas de entrada e de saída. Para seleção do tipo de experimento que se deseja desenvolver, o usuário clica no armário que se abre, apresentando kits referenciados. À seleção do kit, um menu (*widget* 3D) permite que se estratifique ainda mais o experimento desejado.

Por exemplo, ao abrir o armário de Dinâmica, o acesso ao kit de Cinemática permite escolher, por interação direta com o AV, através de menus, entre desenvolver experimentos

de movimentos uniformes ou uniformemente variados. A escolha promove a abertura de uma nova página Web para suportar o desenvolvimento.

Em páginas de elaboração dos experimentos virtuais, objetos de interface - *widgets* - estarão disponíveis para que o usuário possa inserir e retirar elementos, seja na cena propriamente dita, seja no painel de controle, além de definir valores de atributos e armazenar as alterações em arquivos por ele definidos. A inserção de objetos se dá com a seleção do mesmo no kit e arraste do mesmo para a mesa de simulação. A remoção ocorre no sentido inverso.

Assim, ao arrastar um carrinho, por exemplo, para um experimento de movimentação do mesmo (movimento uniformemente acelerado) a partir da aplicação de uma dada força, será necessário definir a massa do mesmo, a força aplicada, o tipo de terreno (para análise de força de atrito) e o tempo de simulação.

A Figura 4.10 apresenta o esquema básico deste tipo de interface, definindo as áreas de objetos de interface, painel de controle e cenário virtual, contendo as três áreas principais desta interface.

A área de *widgets* contém os objetos de interface que propiciam a definição dos objetos que comporão o Cenário Virtual e o Painel de Controle. Na área do Painel de Controle, o autor pode inserir objetos relacionados com o controle dos objetos de cenário e objetos relacionados à medição de variáveis.
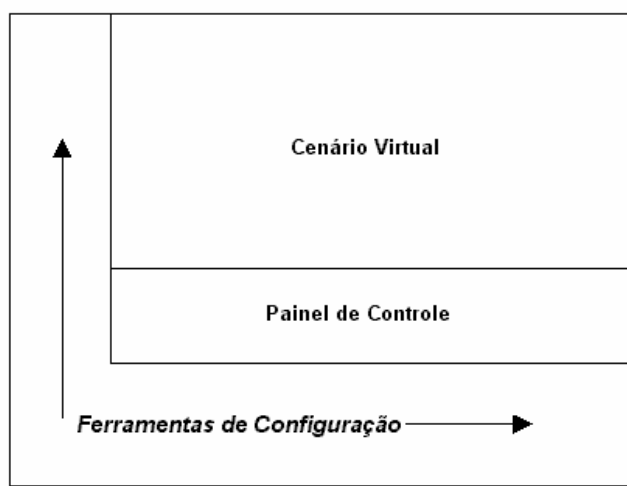
Fig. 4.10 - Esquema Básico da Interface para o Autor

O painel de controle deve conter os objetos relacionados com interações indiretas, de forma a minimizar a perda da imersão subjetiva.

Há, no entanto, possibilidade de o autor optar por inserir menus 3D no Cenário Virtual, de forma a adaptar o sistema para soluções de RV imersivas. Estes objetos de controle são ativados por toques ou seleções de outros objetos existentes no cenário e atuam sobre o objeto chamador atualizando atributos ou ativando métodos. Desta forma, o autor pode optar por soluções de concepção de experimentos que incluem ou não o Painel de Controle.

4.2.2. Interface de Experimentação

A interface será constituída por um cenário virtual (como um laboratório comum) onde armários conterão os kits de experimentos desenvolvidos. São elementos de cenário: cadeiras, mesas, portas de entrada e de saída e kits de laboratório dentro de armários.

Os armários (caracterizados por nomes distintos) conterão os kits arranjados por diferentes tópicos da Física, como Dinâmica, Eletricidade, Óptica e Termologia. Ao clicar sobre o armário, suas portas se abrem e um conjunto de kits é apresentado com referências aos experimentos que representam. Ao selecionar um kit é apresentado ao experimentador uma nova página contendo a simulação desejada. A Figura 4.11 apresenta o esquema básico deste tipo de interface.
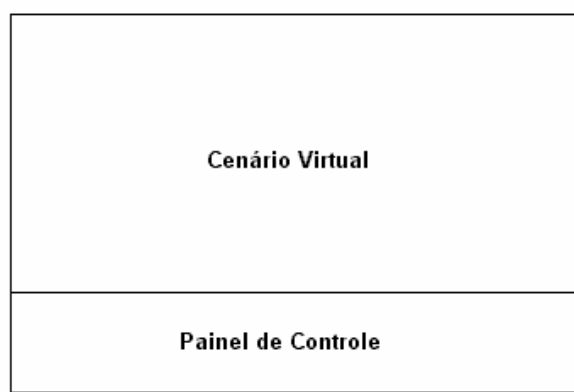
Fig. 4.11 - Esquema Básico da Interface para o Experimentador

Pressupondo que podem haver muitos experimentos relacionados com o mesmo kit, por exemplo, dentro do armário Dinâmica, há diversos experimentos de cinemática, será

permitido ao usuário, quando clica sobre o kit, selecionar, com uso de uma *widget* 3D o desejado. Além disto, uma confirmação será necessária para a abertura da nova página.

O Cenário Virtual contém os objetos 3D que compõem os experimentos com a qual o experimentador interage diretamente. O Painel de Controle contém objetos relacionados com as realimentações para o usuário (*feedbacks*) e outros com a qual o usuário interage (interações indiretas).

É no painel de controle que o usuário pode encontrar *widgets* para alteração de atributos de um dado objeto de forma a promover um novo experimento - comparativo, por exemplo, com os mesmo objetos do cenário virtual em um novo comportamento.

4.2.3. Interface para o Desenvolvedor

A principal característica desta interface é a possibilidade de o usuário modificar o conjunto de objetos do sistema (inclusive, por inserção de novos objetos) que são disponibilizados nos respectivos kits e armários. A interface é diferenciada, visto que este usuário é detentor de conhecimentos mais aprofundados de tópicos de programação. Será necessária uma autorização para que o mesmo possa fazer tais inserções. A função de tais autorizações é evitar que usuários descredenciados desordenem a biblioteca de objetos, levando à necessidade de manutenção do banco de objetos pelos desenvolvedores.

O aplicativo solicita o caminho do arquivo que contém o objeto a ser inserido. Com a confirmação do usuário, abre o arquivo, valida as características do objeto a ser inserido e emite mensagens de erro ou aceitação ao colaborador. Caso existam erros, o aplicativo não se comunica com o servidor do sistema (não transfere o objeto à biblioteca de objetos). Se houver validação, o aplicativo transfere o arquivo correspondente para o servidor (incrementando imediatamente a biblioteca de objetos).

A Figura 4.12 apresenta o esquema básico desta interface. O menu contém *widgets* que permitem acesso a arquivos, edição do código fonte do objeto a ser inserido, acesso às classes do sistema e acesso ao *help* do desenvolvedor. A área central contém o código fonte do objeto que o desenvolvedor está inserindo no sistema e que pode ser comparado com a classe que o conterá. A barra de *status* indica erros no objeto ou problemas de armazenamento relativos à salvar ou carregar arquivos do diretório, que é apresentado na área à direita.
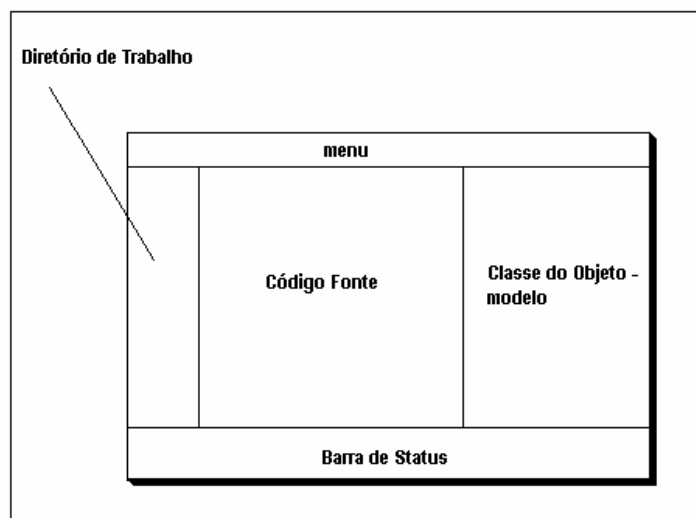
Fig. 4.12 - Esquema da Interface para o Desenvolvedor

Definidas os detalhes da arquitetura, as classes do sistema, as interfaces, pode-se avaliar as proposições através da elaboração de protótipos que correspondam aos subsistemas definidos neste capítulo.

Para o desenvolvimento dos protótipos, será necessário seguir as definições para os objetos dadas pelo diagramas de classes apresentados no Anexo I, compor os módulos dos subsistemas e respeitar o fluxo de informações entre os módulos. O capítulo seguinte apresenta os detalhes de implementação de protótipos do sistema, bem como a avaliação dos mesmos por clientes do sistema.

5. Detalhes de Implementação

Este capítulo apresenta detalhes da implementação dos protótipos desenvolvidos, destacando as ferramentas e tecnologias empregadas, além das principais soluções e dificuldades decorrentes do processo de concepção.

Para fins de desenvolvimento dos protótipos, serão utilizadas as linguagens:

a) HTML, de forma a definir as páginas WEB que serão disponibilizadas, tanto para concepção, quanto para verificação dos experimentos desenvolvidos pelos usuários;
b) VRML, como suporte ao desenvolvimento da biblioteca de objetos e formas 3D que permitam a concepção dos cenários dos experimentos (modelagem geométrica), além da reutilização de código, através de definições de propriedades e atributos destes objetos;
c) JavaScript, integrada a VRML para modelagem dinâmica dos objetos 3D do domínio da solução, e HTML, para concepção de páginas Web;
d) Java, integrada a VRML e HTML através de *applets*, de forma a conduzir o processo de desenvolvimento dos experimentos virtuais, ativando e definindo os objetos que os comporão, além de manusear arquivos e permitir que as modificações dos usuários possam ser armazenadas e reutilizadas. Aplicativos Java, utilizados no subsistema de inserção de objetos, para possibilitar a inclusão de novos objetos ao sistema, de forma a estender a biblioteca de objetos.

As seções seguintes detalham aspectos relevantes destas tecnologias que definiram a utilização das mesmas na implementação de protótipos.

5.1. Linguagens de Programação

5.1.1. VRML

O Apêndice III apresenta um breve histórico da linguagem VRML e exemplos elucidativos referentes às características básicas da mesma.

Arquivos que simulam mundos 3D em VRML, são na verdade uma descrição textual, na forma de textos ASCII, de forma que, por meio de qualquer processador de textos, um desenvolvedor possa conceber tais arquivos. Tais códigos definem como estão as formas geométricas, as cores, as associações, os movimentos, enfim, todos os aspectos relacionados com a idéia do autor (Ames et al., 1997). Quando um dado navegador - *browser* - lê um arquivo com a extensão ".*wrl*" isento de erros, o mesmo constrói o modelo descrito, por ativação de um programa auxiliar específico (*plug-in*).

Um arquivo VRML se caracteriza por quatro elementos principais: - o cabeçalho (obrigatório); - os protótipos; - a descrição da cena: as formas, interpoladores, sensores, scripts e as rotas. O cabeçalho é composto pela instrução "#VRML V2.0 utf8" e sua omissão impossibilita o *plug-in* do navegador de ler o arquivo em questão.

Destacam-se algumas características de VRML que são relevantes a este trabalho:

(a) Usando VRML, a concepção de cenários tridimensionais se baseia na elaboração de uma grafo direcionado acíclico, contendo diferentes ramos (ou nós), que, associados de forma correta podem ser agrupados. A grande diversidade destes nós (cinqüenta e quatro pré-definidos), -incluindo primitivas geométricas, propriedades de aparência, sons (e propriedades) e vários tipos de nós de agrupamentos, é uma das principais características, e qualidades, da linguagem.

(b) VRML permite reutilização de código através da prototipação, baseada na definição de novos nós, denominados *protos*, que podem ser definidos em um arquivo, na forma de objetos e posteriormente, serem utilizados por outros arquivos ou ativados dentro de um arquivo como um nó externo, evitando a duplicação de códigos.

(c) Do ponto de vista de construção de cenários 3D, VRML provê nós para criação e modificação de fundos de cenários, com o uso de nós específicos - *backgrounds*, - que simulam ambientes diferenciados assemelhados a condições que variam de um dia ensolarado, para nublado, com neblina ou mesmo noites.

(d) VRML possibilita controle de aparência de elementos do cenário e a inserção de diferentes formas de fontes de luz (pontuais, direcionais, ambiente), o que permite dar mais realismo ao cenário concebido. Recursos de acrescentar sons e filmes também estão disponíveis por utilização de nós específicos e são compatíveis com os principais formatos de áudio e vídeo: .mpeg, .mpg, .mid., .wav (Allison, 1997).

(e) Associada a *Java* ou *JavaScript*, podem ser elaborados *scripts*, de forma a complementar a troca de informações entre os elementos do mundo virtual e destes com os usuários. Esta propriedade provê possibilidade de animações e de dinamismo às formas concebidas e inseridas no cenário.

5.1.2. JAVA

O Apêndice III apresenta histórico da linguagem e exemplos de aplicações.

Java é uma linguagem de programação capaz de permitir páginas altamente interativas na WEB, orientada a objetos, que foi modelada de acordo com a linguagem C++ e projetada para ser simples. Segundo Chan et al. (1999); Ritchey (1997), são principais características de Java:

(a) Portabilidade: O código fonte de programas em Java é pré-compilado em *bytecodes*, que são instruções semelhantes ao código de máquina e não são específicos para plataformas (processadores). Assim, qualquer máquina que possua o interpretador Java pode executar tal código. Praticamente, todos os navegadores compatíveis com a Internet já incorporam as máquinas virtuais Java - *JVM: Java Virtual Machine*;

(b) Robustez: O código deve se comportar adequadamente, primariamente por portabilidade e secundariamente pelos recursos da linguagem e do ambiente;

(c) Segurança: o ambiente Java não permite ações premeditadas no sentido de executar programas nocivos na máquina cliente. *Applets Java* não podem executar entradas e saídas em disco e não enxergam a distribuição de memória do computador;

(d) Orientação a objetos: permite herança e reutilização de código de forma dinâmica e estática. Para Java, uma classe é um conjunto de dados e métodos que descrevem um objeto com a qual um dado programa trabalha. As classes podem ser construídas de modo hierárquico, sendo permitida a criação de novas classes que podem ser reutilizadas;

(e) Dinamismo: Java é dinâmica por sua própria natureza e permite a extensibilidade durante a execução;

(f) Alto desempenho: permite *multithreading*, compilação *just-in-time* e utilização de código nativo;

(g) Facilidade: o ambiente é familiar a programadores de C e C++, além de retirar do programador a responsabilidade por tarefas sujeitas a erro, tais como gerenciamento de memória e de ponteiros.

O desenvolvimento de programas - aplicativos e *applets* - usando Java pode ser feito em diversos ambientes, sendo a maior parte deles do tipo *freeware*. A Sun disponibiliza o kit do desenvolvedor Java - *JDK*, que fornece todas as ferramentas básicas para compilar e executar os códigos em Java (JAVA, 2000). A *Borland* disponibiliza um ambiente de desenvolvimento compatível com os sistemas operacionais *Windows*® e *Linux*, o Jbuilder. A Microsoft apresenta o VisualJ++ compatível com a família *Windows*.

5.1.3. JavaScript

A despeito da semelhança de nomes, JavaScript não tem relação com a linguagem Java do ponto de vista conceitual. Enquanto Java é compilada, JavaScript é uma linguagem interpretada (JavaScript, 2001), voltada ao desenvolvimento de programas pequenos (*scripts*). O código JavaScript é lido pelo navegador como código-fonte (Winters et al., 1997).

Scripts podem se associados ao código HTML de uma página Web, propiciando uma forma acessível de aumento de interatividade e funcionalidade, ou serem inseridos junto ao código ASCII correspondente a um arquivo VRML (Ames et al., 1997), permitindo alterações dinâmicas dos objetos do mundo virtual modelado, como o controle de atributos relativos a posição e aparência.

Associada a VRML, JavaScript permite a modelagem comportamental dos objetos, provendo animações mais complexas que as permitidas pelos nós originais da linguagem. Este *script* é tratado como componente de um nó e seu código deve ser escrito dentro do mesmo. Tipicamente, tais programas permitem ações mais complexas, associadas com: resultados de saída de animações computacionais; trocas de informações entre computadores da rede ou a definição de comportamentos baseados nas ações dos usuários e dos demais objetos do mundo modelado.

Compõem esta forma de *script*:

- Dados comuns: relativos a valores de variáveis manuseados pelas funções;
- interfaces de entrada de dados: associadas ao recebimento de mensagens de objetos do mundo virtual;

- interfaces de saída de dados: relativas ao envio de dado a objetos ou *scripts* do mundo virtual;
- corpo de funções: contém as funções responsáveis pelo manuseio dos dados recebidos como eventos externos e pelo envio de dados.

Através desta associação, é possível a concepção de objetos VRML capazes de respostas a solicitações externas e alteração de atributos dinamicamente.

5.2. Concepção de Objetos e suas Propriedades Físicas

Definidas as tecnologias de software utilizadas no projeto, faz-se a modelagem dos objetos através da definição de seus comportamentos e interações. São modelados alguns componentes para a biblioteca de objetos, de forma a viabilizar a elaboração dos protótipos das fases de configuração e teste do sistema.

Durante esta etapa, são modelados:

- objetos do meta-ambiente;
- objetos atuadores e sensores;
- objetos de experimentos;
- objetos de adorno;
- menus, sensores e atuadores 3D.

Na fase de modelagem, serão necessárias simulações de propriedades físicas dos elementos que são inseridos nos protótipos. De forma a simular propriedades físicas dos objetos de um mundo virtual, métodos correspondentes a comportamentos físicos, descritos por *scripts* JavaScript são indispensáveis (Cardoso et al., 1999). A inserção de códigos JavaScript dentro de um arquivo que define elementos 3D em VRML, pode ser feita através do nó *Script* (Ames et al., 1997), (Lawton, 1999) e da correta associação de eventos de entrada (*eventIn*) e de saída (*eventOut*).

Por outro lado, de forma a implementar as interfaces de configuração e do desenvolvedor, faz-se uso da linguagem Java. Java será inserida em protótipos visando:

- prover o acesso a arquivos que contêm objetos ou experimentos;
- permitir a interação com os ambientes virtuais (através de *applets*);
- o desenvolvimento de aplicativos independentes.

As seções seguintes destacam aspectos de implementação de objetos, que compõem os protótipos desenvolvidos. São realçados itens relevantes relacionados às características

das implementações do modelo geométrico e comportamental dos mesmos e a utilização das linguagens de programação relacionadas.

5.2.1. Objetos de Adorno: Fundos de Cenas

Artifícios de redução do campo visual, de forma a definir um aspecto importante de um cenário virtual podem ser opressivos para o usuário (Ames et al., 1997), mas, há momentos em que tais estratégias são necessárias. No entanto, quando é necessário permitir uma visão não obstruída, incluindo um cenário mais elaborado dotado de uma ambientação que inclua o céu do mundo virtual e o horizonte distante, faz-se necessária a construção e utilização dos fundos de cenas.

A linguagem VRML provê nós relativos a tais fundos de cenas, como o nó *background*. Para flexibilizar a inserção destes, os protótipos devem permitir a escolha do mais adequado pelo autor, no momento da concepção do A.V., além da definição de novos fundos de cenas pelo desenvolvedor.

A Figura 5.1 apresenta a definição da classe "Fundo de Cena" e objetos relativos a esta classe disponíveis para o autor, que foram denominados: "Básico", "Gramado" e "Praia".
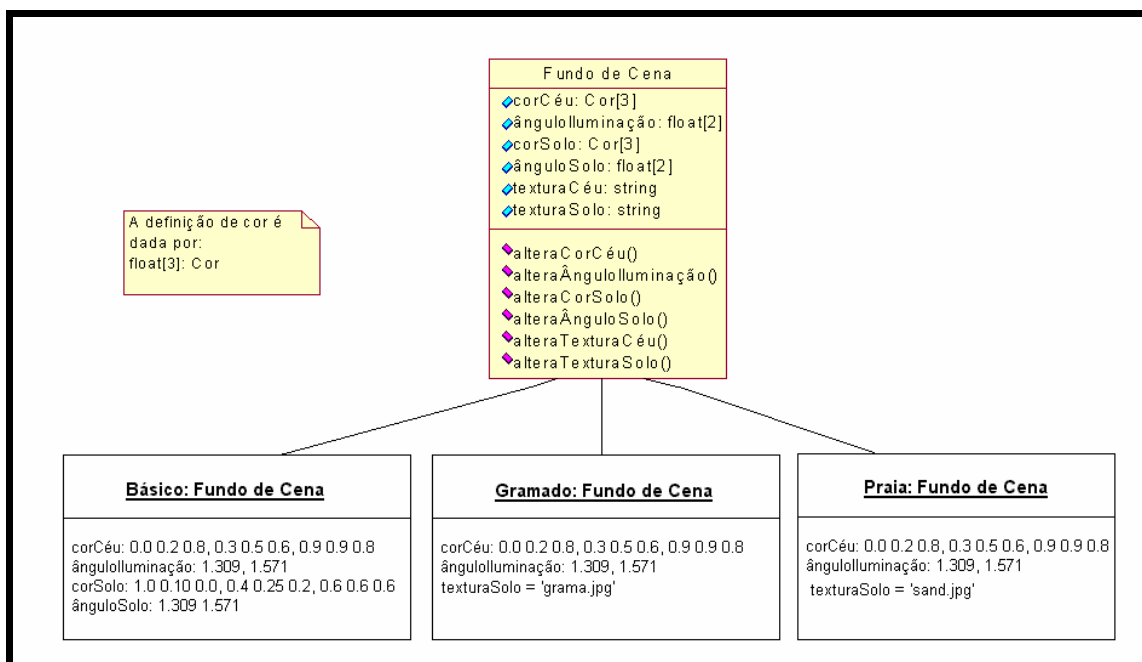
Fig. 5.1: Objetos relativos à classe Fundo de Cena

Observa-se que os atributos da classe "Fundo de Cena" são relativos ao céu e solo que será visualizado no cenário virtual e que as operações permitem a alteração de tais atributos a partir de mensagens externas que contenham parâmetros de modificação, se necessário.

Os objetos destacados na Figura 5.2 relacionam-se com panos de fundo característicos de um dia ensolarado, com solos variando entre um solo comum, sem qualquer característica particular e solos característicos de um gramado ou da areia de uma praia. As texturas aplicadas anulam as características definidas pelos atributos "corSolo" e "ânguloSolo", que só têm validade na ausência total de texturas.

A Figura 5.2 apresenta os resultados de implementação dos três objetos citados.
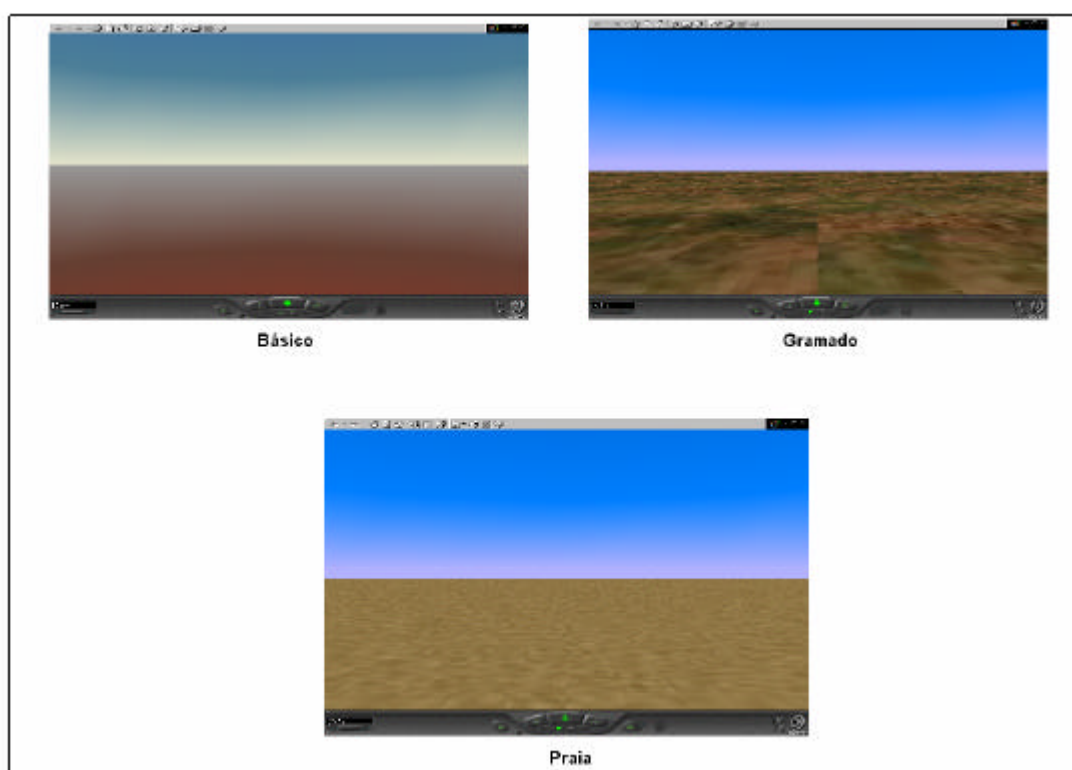
Fig. 5.2 - Resultado da Implementação dos Objetos Relativos à "Pano de Fundo"

5.2.2. Objetos de Adorno: móveis e complementos

Móveis são implementados para dar mais realismo ao cenário virtual e permitem decorar o ambiente modelado de forma adequada à abstração de um dado cenário. Um laboratório, por exemplo, pode possuir pranchetas, lousas, mesas, cadeiras, computadores,

quadros decorativo, dentre outros. Por meio da utilização de tais objetos, o usuário em seu acesso ao sistema, tem melhorada a sensação de imersão no ambiente.

A Figura 5.3 apresenta a definição do objeto cadeira, relativo à classe "Objetos de Adorno".
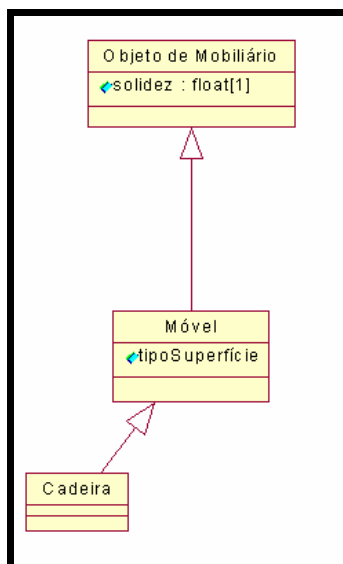
Fig. 5.3. Definição de um Objeto Cadeira

Dentre os Objetos de Adorno há ainda outras classes relativas a mesas, portas, suportes etc. A Figura 5.4 apresenta o resultado da implementação da cadeira, supondo atributos de cor relativos à cor vermelha.

Fig. 5.4 - Objeto Cadeira

5.2.3. Objetos de Experimentos: Molas

Com o propósito de criar um pacote de objetos que permita ao usuário inserir, remover, definir atributos, reutilizar e adequar os mesmos às suas necessidades, são definidos objetos que simulam elementos físicos reais, como molas, carrinhos, roldanas (polias), massas etc.

Para definir estes objetos, permitindo reutilização, a implementação utiliza nós do tipo *proto* da linguagem VRML, que permitem estender a biblioteca de formas do VRML, uma vez que um usuário pode definir seu próprio nó (Ames et al., 1997).

Na definição dos objetos de experimentos este recurso é relevante, pois possibilita que seja definido um nó (relacionado a um objeto), valores relativos ao modelo implementado (relativos aos atributos), as operações sobre estes valores (relativas aos métodos) e os campos de entrada e saída de valores relacionados com o nó (mensagens oriundas do objeto ou por ele recebidas). A Figura 5.5 representa a adequação de utilização dos nós *proto* para a definição dos objetos de experimentos do sistema, através da associação de componentes das classes com componentes disponíveis nos nós *proto* da linguagem de implementação.
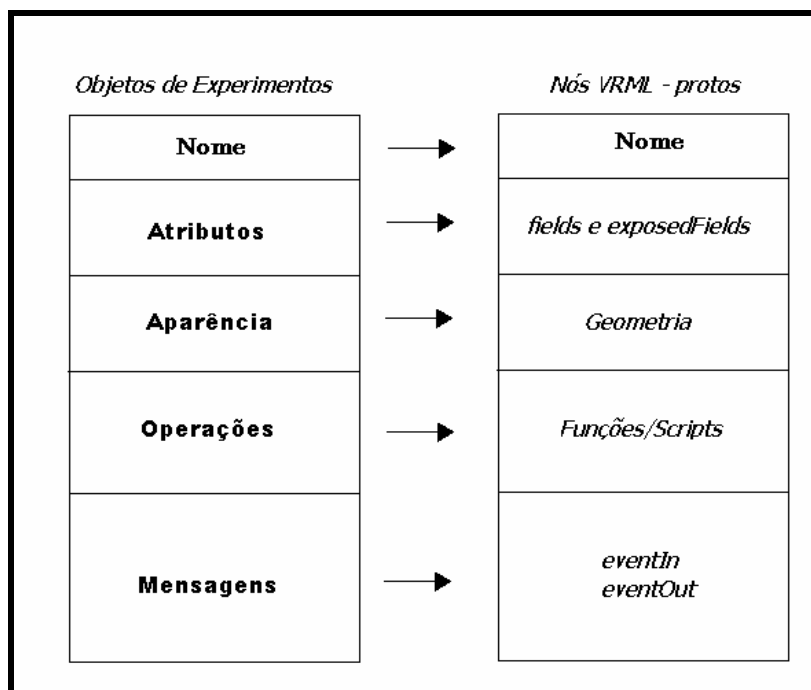
Fig. 5.5 - Relação de Objetos de Experimentos com nós VRML

Os campos do nó *"proto"* relacionados com a interface do nó (*fields*, *exposedFields*, *eventIn*, *eventOut*) podem ser do tipo real, inteiro, *string*, booleano e relativos a tempo. A descrição da aparência do objeto está relacionada com a geometria do nó, podendo ser feita com a utilização de nós existentes ou definidos pelo usuário (outros nós *"proto"*). Funções e *Script* descritas no nó *"proto"* operam sobre os atributos e definem as entradas e saídas de mensagens.

Como exemplo, com a utilização de um objeto Mola, podem ser feitas simulações relativas à lei de *Hooke* ou de conversão da Energia Mecânica (Potencial em Cinética). Considerando a implementação, é necessário modelar geometricamente a mola e definir o conjunto de atributos e operações, para sua inserção no pacote de Objetos de Experimentos.

A classe mola pertence à superclasse "Objetos de Experimentos" e à classe "Objeto Elástico" e contém a modelagem geométrica e comportamental que acomoda a especificação de um objeto elástico caracterizado por uma constante elástica (*k*).

A Figura 5.6 apresenta a definição da classe Mola, destacando atributos específicos como: número de espiras (*nEspiras*), raio da seção transversal da mola (*raioSeção*), comprimento (*comprimento*) e espinha (*spine*, de visibilidade privada).
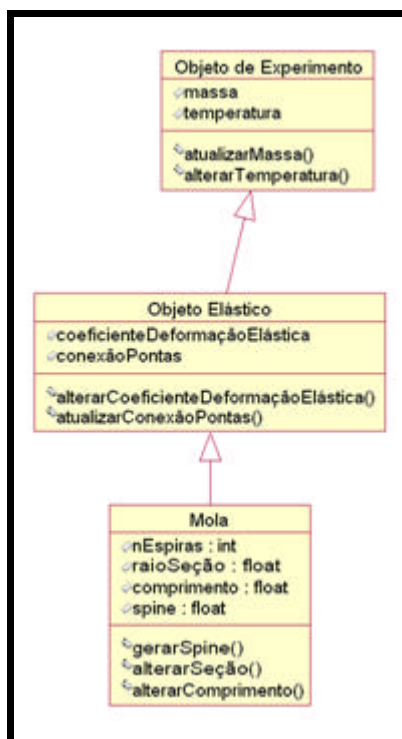
Fig. 5.6 - Definição da Classe Mola

As operações de construção e de comportamento do objeto são: *gerarSpine*, para gerar a espinha de extrusão, *alterarRaioSeção*, usada para afinar a mola em caso de grande alongamento e *alterarComprimento*, capaz de definir o comprimento do objeto no cenário virtual.

Considerando a modelagem geométrica, adotou-se a solução de promover a extrusão de uma forma circular de raio de Seção igual a *raioSeção* em torno de uma hélice definida pelos atributos *spine, nEspiras* e *comprimento* conforme ilustra a Figura 5.7.

Atributos como a cor da mola são associados com elementos da geometria, como o campo *diffuseColor,* que está inserido dentro do nó *appearance* do código fonte em VRML. A criação da espinha que contém a linha básica se dá com uso do nó *extrusion* que suporta tal modelagem.
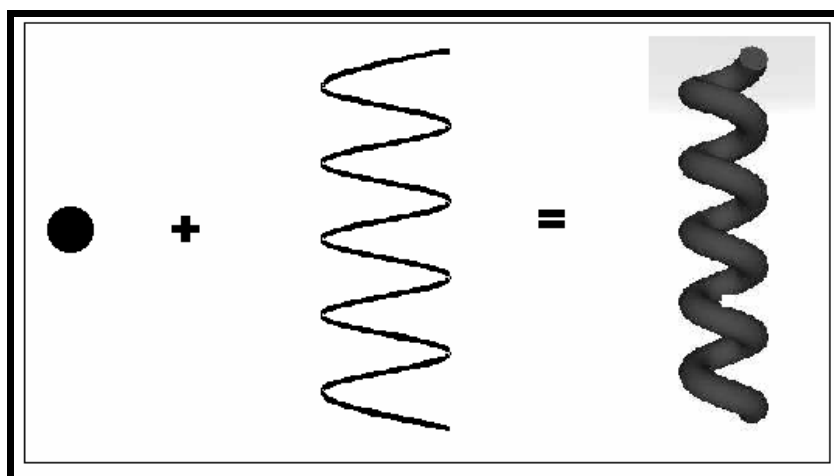
Fig. 5.7 - Definição Geométrica do Objeto Mola

Do ponto de vista comportamental, é necessário permitir que esta mola se deforme e possa oscilar (segundo a lei de *Hooke*). Para prover tal solução, uma função JavaScript permite reconstruir a mola, modificando a espinha sempre que necessário. Um conjunto de funções definidas dentro de um nó *script* descrevem estas operações e contém entradas relacionadas com os atributos e com as mensagens que podem ser recebidas ou enviadas pelo objeto.

Como nova mensagem é permitido o recebimento de um novo valor de alongamento para a mola (distância entre duas espiras consecutivos). Neste caso, é ativada uma função que calcula a nova espinha e a envia para a função modeladora. Esta troca de informações

entre as funções que definem um novo alongamento ou um comportamento dinâmico da mola.

5.2.4. Objetos de Experimentos: Carrinhos

Além de objetos elásticos, representados na seção anterior pela mola, os protótipos apresentam definições relativas a objetos rígidos. Estes são utilizados em experimentos relacionados com a cinemática, óptica geométrica, eletricidade etc.

Um desses objetos é o carrinho, muito comum em laboratórios para uso em experimentos de comprovação de equações da cinemática e validação de princípios da dinâmica, como as leis de Newton.

Para definição deste objeto, devem ser considerados atributos como: velocidade escalar, velocidade angular, coeficiente de atrito, além dos atributos herdados da classe "Objetos de Experimento", como a massa. A Figura 5.8 apresenta a estrutura da definição da classe "Objeto Rígido".
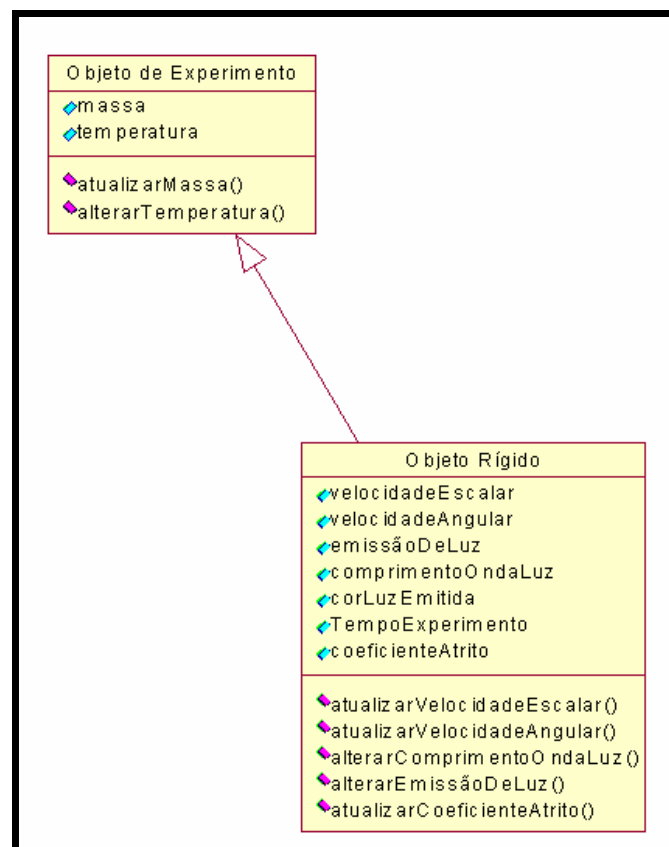
Fig. 5.8 - Classe Objeto Rígido

A Figura 5.9 apresenta o resultado da implementação de um objeto *Carrinho*, que tem métodos para atualizar a velocidade escalar e angular, além do coeficiente de atrito. Tal objeto está disponível nos kits de experimentos relativos à Cinemática e Dinâmica e pode ser usado pelo autor, por exemplo, para ilustrar uma aula da 2ª lei de Newton.

Uma das características deste objeto é a capacidade de ativar um menu3D a partir de um clique sobre o mesmo. Para implementar tal opção, o objeto dispõe de operações que geram mensagens para o objeto Menu3D, ativando-o e recebendo mensagens do mesmo, para atualização de atributos.
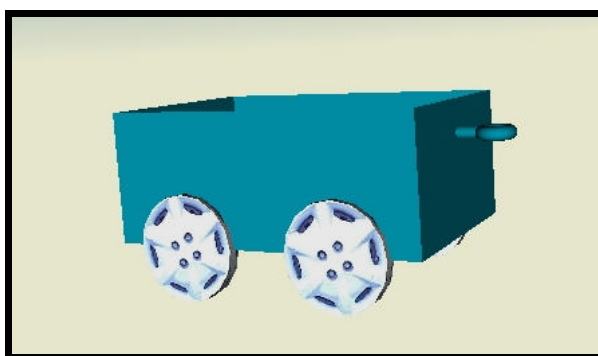
Fig. 5.9 - Carrinho de Laboratório

A Figura 5.10 apresenta o objeto e um menu 3D ativado ao clique do usuário sobre o objeto.
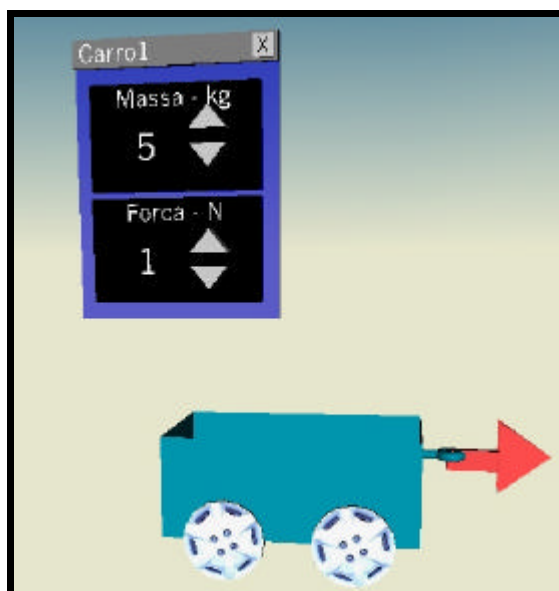
Fig. 5.10 - Objetos Carrinho e Menu associados

5.2.5. Objetos de Interface

Objetos de interface (*widgets*) são responsáveis pela troca de mensagens entre objetos dos Ambientes Virtuais e usuário. A implementação de tais objetos devem prover a possibilidade de modificação da visualização objeto de cenário com quem trocam mensagens. Assim, um mesmo botão ou menu que altera o valor da massa de um móvel em experimento de Movimentos Acelerados, pode ser usado para mudar uma constante elástica de uma mola ou a dimensão de um objeto que está defronte a um espelho.

Para estes fins, dois tipos diferentes de objetos de interface foram implementados: objetos de interface 3D (que participam do cenário virtual, permitindo uma interação direta, e objetos de interface 2D (que compõem o painel de controle), para suportar a interação indireta. O próprio painel de controle é um objeto de interface capaz de agrupar outros objetos, como botões de controle e caixas de texto.

Como objetos de interface 3D temos alguns elementos representados no Cenário Virtual, como móveis, que podem ser utilizados para permitir ao usuário ativar experimentos, arrastar objetos e outras interações diretas.

Um exemplo deste tipo de aplicação é dado pelo gaveteiro: um conjunto de gavetas capaz de ativar a chamada da interface de experimentação ou o meta-ambiente para a criação de um novo experimento relativo a um dado kit de experimentos, associado com o *label* da gaveta selecionada.

O resultado da implementação de um objeto desta classe é apresentado na Figura 5.11. Cada gaveta recebe um rótulo que permite uma ativação específica. Quando o usuário clica sobre a gaveta escolhida, o objeto envia uma mensagem para o navegador, ativando a carga de uma nova página relativa ao link correspondente à tal seleção.

Há ainda dois outros tipos de objetos disponíveis para interação direta: o menu3D e o controle deslizante. O menu3D permite a atualização discreta de valores de dois atributos numéricos de um objeto controlado a partir da definição de máximo, mínimo e incremento.
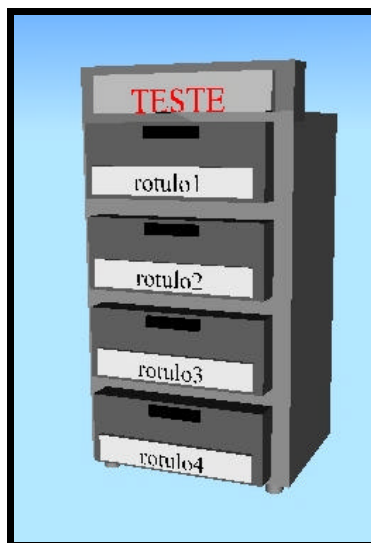
Fig. 5.11 - Objeto Gaveteiro

A Figura 5.12 apresenta um exemplo deste menu, que também é implementado como um *proto* em VRML e contém funções JavaScript que permitem a definição posterior dos atributos: Nome do Menu, Valores Máximos, Mínimos, Incrementos, Cores e se, inicialmente, estão ativos ou não.

Fig. 5.12 - Menu (Atuador) 3D

Este menu pode ser escondido pelo clique no X acima à direita. A ativação dos mesmos é feita por mensagem oriunda de outro objeto do cenário virtual, que pode ser um objeto de adorno, experimento ou de interface. Variações deste menu podem ser utilizadas para chamar páginas Web.

O controle deslizante, ao contrário - pode ser utilizado para variar continuamente um dado valor numérico em uma faixa definida pelo usuário. A Figura 5.13 ilustra um objeto Controle Deslizante, para o qual foi dado o nome "C D".
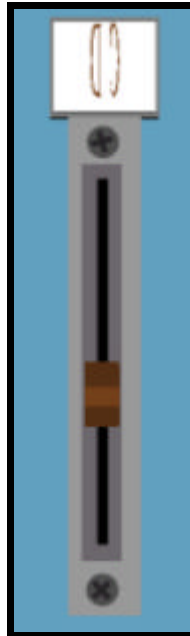
Fig. 5.13 - Objeto Controle Deslizante

5.3. Concepção dos Subsistemas da Arquitetura

5.3.1. Introdução

No desenvolvimento dos protótipos do sistema, a opção pela união de HTML, VRML, Java e JavaScript dificulta a utilização dos geradores de páginas Web comumente usados, tais como o Netscape Composer, Microsoft FrontPage, DreamWeaver. Com o propósito de concepção dos protótipos dos subsistemas e das interfaces, o desenvolvimento destas é feito com uso de editores de texto comuns e programas de concepção de *applets* e aplicativos Java.

Visando implementar os protótipos compatíveis dotados de interfaces compatíveis com a Web, as seguintes associações são utilizadas:

(a) HTML e VRML para o desenvolvimento das páginas Web que contêm cenários virtuais do sistema;

(b) HTML e JavaScript para desenvolvimento de páginas compostas por *frames* e para definição de objetos de interface;

(c) VRML e JavaScript na concepção de objetos do sistema (modelagem comportamental), além de alguns objetos de interface para prover ao autor e ao experimentador a seleção do experimento, abstraindo o comportamento de escolha de um kit em um laboratório real;

(d) HTML, VRML e *applets* Java para prover *widgets* que suportam a definição dos objetos e suas relações pelo autor, na definição de um A.V. referente a um dado experimento. Tais *applets* interagem com mundos virtuais que estão sendo definidos de forma dinâmica propiciando a imediata atualização do cenário e do painel de controle, permitindo ao autor a modificação de atributos dos objetos inseridos e o teste das operações de troca de mensagens entre estes objetos.

5.3.2. Subsistema de Criação de Experimentos

O subsistema de criação de experimentos é caracterizado por uma interface de constituída de um conjunto de páginas Web, que permite ao autor definir um cenário virtual, um painel de controle, os atributos dos objetos inseridos e os relacionamento destes (através das trocas de mensagens).

Pelo acesso ao sistema, o autor opta por criar um novo experimento e escolhe um subdomínio da Física. Tal escolha define as páginas Web que serão apresentadas ao autor de forma a permitir que o mesmo desenvolva a criação de um experimento.

Conforme a seleção do usuário, uma página específica de desenvolvimento é carregada. Esta página é constituída de um *applet* Java e de um cenário virtual (concebido em VRML), de tal forma que objetos do *applet* atuem sobre elementos de cenário virtual. Como solução de implementação, faz-se utilização da *External Authoring Interface - EAI*, que associa o *plug-in* VRML à *Java Virtual Machine* do navegador e permite a comunicação bidirecional entre o *applet* e o *plug-in* VRML (Marrin, 2001).

A Figura 5.14 apresenta a associação entre VRML, Java e a EAI utilizada neste arranjo (Cardoso et al., 2001).
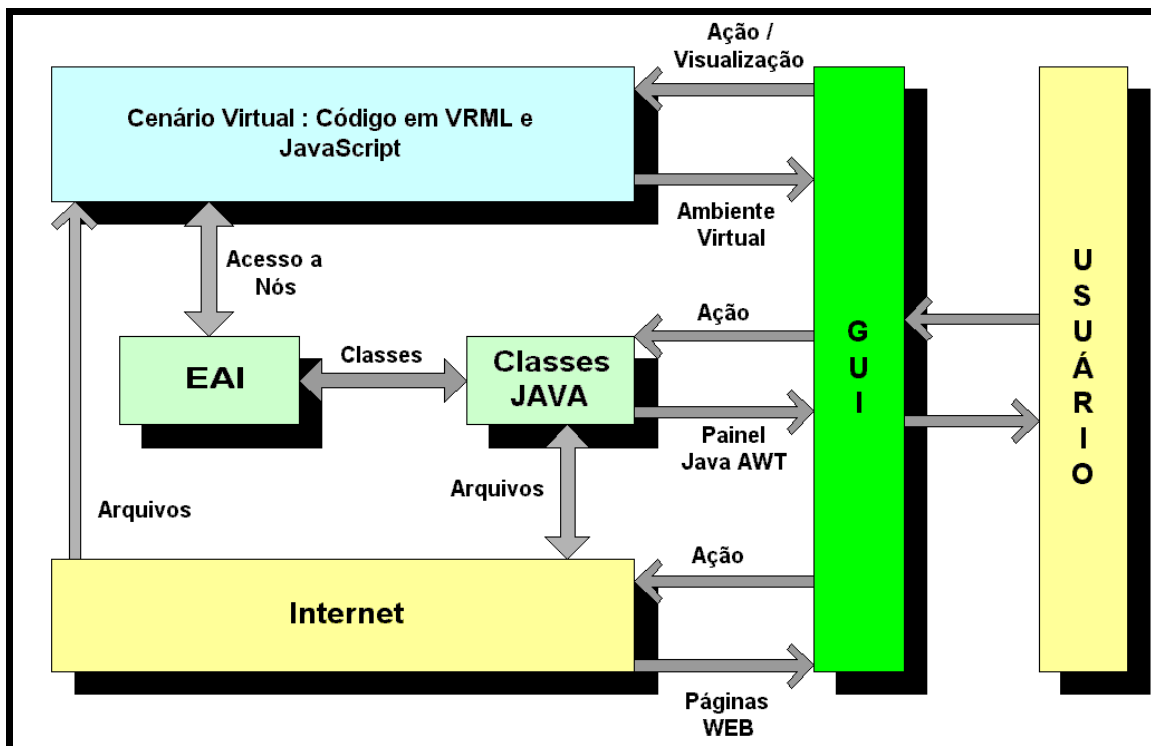
Fig. 5.14 - Esquema de Comunicação EAI, VRML, Java.

Com uso desta estratégia de implementação, cabe ao *plug-in* VRML a tarefa de representar a cena tridimensional e ao *applet* Java as tarefas de:

(a) definir que elementos serão inseridos no cenário virtual;

(b) definir posicionamento dos elementos que compõem a cena 3D, informando ao *plug-in* qual a posição (x,y,z) de cada um no cenário;

(c) permitir que o usuário mude as propriedades do objeto de experimento que está sendo inserido, como massa, força aplicada, tempo de amostragem etc, através de campos de texto onde o usuário insere as informações solicitadas;

(d) definir as relações entre objetos que estão no ambiente virtual. Por exemplo, caso o usuário insira um botão de controle 3D, definir a que elementos tal botão está associado;

(e) atualizar as informações dadas pelo criador de experimento num *buffer* que conterá o arquivo com todas as informações do cenário;

(f) verificar erros relativos a entradas inválidas de dados pelo autor;

(g) manusear o arquivo que conterá as informações do *buffer* e que poderá ser recuperado posteriormente, através de chamadas, no servidor de arquivos.

Os itens (e), (f) e (g) da lista anterior demonstram que o *applet* executa o papel de configurador do ambiente virtual, verificador de erros, contendo um gerador de códigos que é capaz de produzir o código relativo aos componentes criados pelo autor e um tratador de arquivos, todos relativos à arquitetura do Subsistema de Criação de Experimentos.

A Figura 5.15 representa uma tela típica da GUI (existente em protótipo desenvolvido) relativa à Criação de Experimentos.
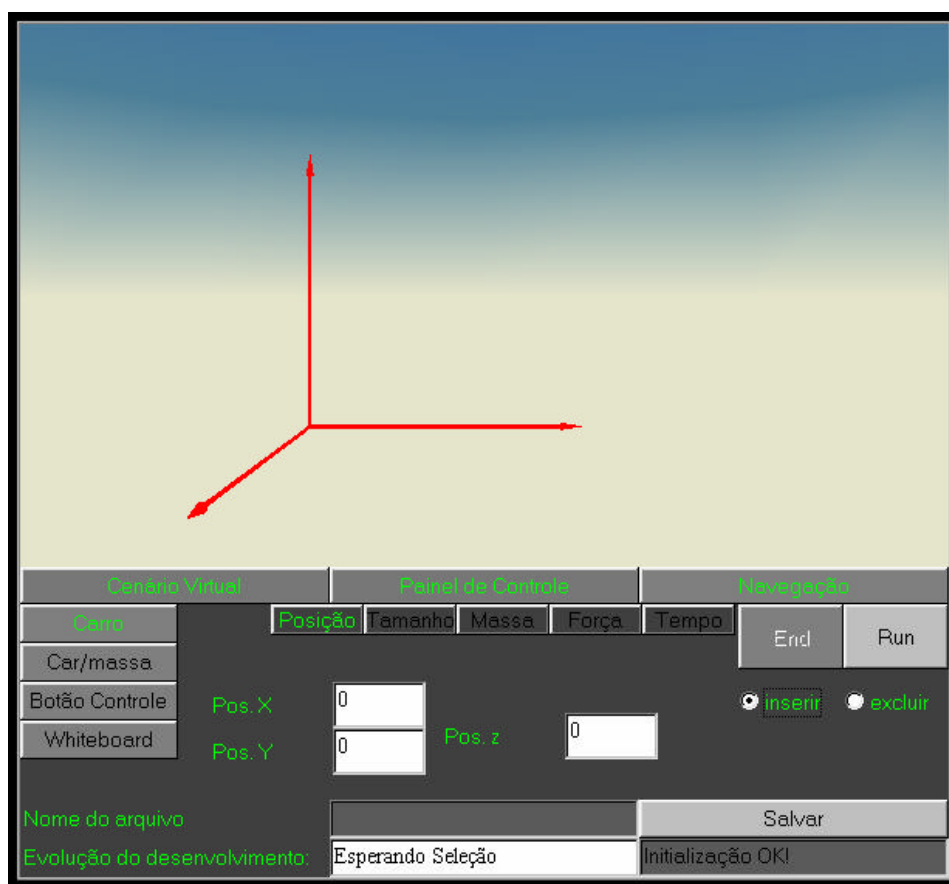
Fig. 5.15 - Interface de Criação de Experimentos: Cinemática

Na concepção da GUI do Subsistema de Criação de Experimentos, a integração da linguagem Java com HTML e com o *plug-in* VRML foi necessária. O protótipo desta interface se caracteriza por permitir que as definições feitas no *applet* sejam imediatamente apresentadas no cenário virtual e armazenadas pelo *applet*, para posterior salvamento.

De forma a definir os componentes do cenário virtual, o *applet* mantém um gerador de códigos que, baseado nas definições do autor, vai construindo um texto relativo aos objetos que comporão o cenário virtual. Tal gerador de códigos VRML consulta uma tabela de códigos dos objetos e, baseado nos códigos fundamentais, altera-os e armazena temporariamente uma *string*, que depois é repassada para o *plug-in* para a representação no cenário. A Figura 5.16 apresenta um fragmento do código do *applet*, relativo à captura e validação dos nós do arquivo VRML inicial, que serão modificados dinamicamente (pelas definições do autor).

```
public void initScene() {  try {
        Adiciona = browser.getNode("Adiciona");
                        grupo   = browser.getNode("Grupo");
                        nav2    = browser.getNode("nav2");
                        nav     = browser.getNode("nav");
                        Principal = browser.getNode("Principal");
                        warningText = browser.getNode("warningText");
                        vista = browser.getNode("Vista");
                        output.setText("Initialização OK!");
        }
        catch(InvalidNodeException e){
        output.setText("PROBLEMS!: " + e );
        return;
        }
        try {           IsClicked = (EventInSFString)Adiciona.getEventIn("IsClicked");
                        RemClicked = (EventInSFString)Adiciona.getEventIn("RemClicked");
                        set_whichChoice = (EventInSFInt32)Grupo.getEventIn("set_whichChoice");
                        LerString = (EventInSFString)Adiciona.getEventIn("LerString");
                        texto.setText("Escolha o Fundo do Cenário!");
                        set_bind = (EventInSFBool)nav2.getEventIn("set_bind");
                        set_bind2 = (EventInSFBool)nav.getEventIn("set_bind");
                        set_position = (EventInSFVec3f)Vista.getEventIn("set_position");
                        addChildren1 = (EventInMFNode) Principal.getEventIn("addChildren");
                        removeChildren1 = (EventInMFNode) Principal.getEventIn("removeChildren");
                set_whichChoice_wt=(EventInSFInt32)warningText.getEventIn("set_whichChoice");
        }
```

Fig. 5.16 - Fragmento de Código do *applet* para Captura dos nós VRML

As instruções do tipo "browser.getNode" iniciam a troca de informações entre o applet e o *plug-in* VRML que deve estar apresentando um arquivo VRML que contenha os nós principais (inicialmente sem outros nós agrupados). A análise da figura permite identificar que o protótipo está relacionado com um arquivo VRML que tem como nós principais os nós "Grupo", "nav2", "nav", "Principal", "warningText" e "Vista". A ausência de qualquer um destes nós no arquivo carregado pelo navegador VRML é razão de erro.

No processo de criação, o autor define os componentes do cenário virtual e seus atributos. Para cada componente definido, uma imediata representação do mesmo é provida no cenário virtual do protótipo, de forma a permitir a visualização da adequação do mesmo aos propósitos do autor. A solução de protótipo relativa ao configurador de ambientes virtuais fundamenta-se em armazenar as alterações do autor em uma *string*, verificar os erros e passar informações para o *plug-in* apresentar no cenário virtual. A Figura 5.17 refere-se ao fragmento de código da respectiva classe Java que envia ao *plug-in* a informação do objeto virtual a ser representado.

```
try     {
        initScene();
        no_int = browser.createVrmlFromString(Stringval);
                if(escolha == "inserir"){
                        addChildren1.setValue(no_int);}
                if(escolha == "excluir") {
                        texto.setText("Tentando Remover");
                        varia = browser.getWorldURL();
                        String[] variavel = null;
                        removeChildren1.setValue(no_int);
                                }
        }
        catch (InvalidNodeException e) {
                output.setText("PROBLEMS!: " + e + "\n");
        }
```

Fig. 5.17 - Fragmento de código que envia dados para o *plug-in* VRML

Distingue-se como principal instrução deste fragmento de código a linha "browser.createVrmlFromString", que define ao *plug-in* que represente um dado objeto relativo à *string* denominada "Stringval", que, neste caso em particular representa o objeto

a ser inserido no cenário virtual. Tal objeto é, simultaneamente, armazenado em uma lista dentro do *applet*. Há listas para os diferentes tipos de objetos possíveis, que os agrupa por natureza. Assim, objetos rígidos estão em uma lista diferente de objetos deformáveis etc. Tal estratégia permite que as diferentes associações sejam construídas e validadas.

Quando o autor define por inserir um quadro explicativo no cenário virtual, o *applet* executa o papel de analisar e quebrar o texto em diversos elementos léxicos (*tokens*) de forma a formatar o texto para inserção no quadro. A Figura 5.18 refere-se ao fragmento de código que executa tal tratamento.

```
valorz = textoz.getText();
valorq = textoquadro.getText();
    if (valorq.length() != 0){
            int num_iter;
            int num_token;
            num_iter = 0;
            String token;
            int index;
            int contar;
            StringTokenizer tokens = new StringTokenizer(valorq," ");
                num_token = tokens.countTokens();
                token = "";
                token = tokens.nextToken();
                textotemp.insertElementAt(token,0);
                while(tokens.hasMoreTokens()){
                    Stringint = Stringint  + " " + "\"";
                    contar = 0;
                    StringBuffer temporaria = new StringBuffer(" ");
                    while(contar < 18){
                    temporaria.append(" ");
                        if( tokens.hasMoreTokens()){
                            token = tokens.nextToken();
                            temporaria.append(token);
                            texto.setText(temporaria.toString());
                            contar += temporaria.length();
                            }
                            else{break;}
                            }
                Stringint = Stringint + temporaria.toString();
                Stringint = Stringint + "\"";
                Stringint = Stringint + " ";
                } ...
```

Fig. 5.18 - Fragmento de Texto para tratamento de *tokens*

Como o quadro tem uma extensão de 18 caracteres, os tokens são divididos e quebrados de forma a inserção formatada dentro do quadro explicativo.

O *applet* também é responsável pelo tratamento de arquivos, que conterão a descrição do ambiente virtual criado pelo autor. A Figura 5.19 apresenta o fragmento de código que abre um arquivo, armazena as informações e salva o mesmo no servidor de arquivos.

```
if(evt.target == botaon){
        String strn;
        strn = texton.getText();
        estado = strn;
        String arq;
        boolean control2;
        control2 = true;
        if( control2){
                try{
                diretorio = "c:" + "\\" + "arq_v~1" + "\\";
                RandomAccessFile file = new RandomAccessFile(diretorio + strn +".wrl","rw");
                file.writeBytes("#VRML V2.0 utf8");
                file.writeBytes("\r");

                                    ...

                file.writeBytes(Strrotas);
                file.close();
                texto.setText("Arquivo salvo - Fim de operação");
```

Fig. 5.19 - Fragmento de código do Tratador de Arquivos - *applet* Java

Ao final da operação "*Salvar*", uma mensagem de operação completa é mostrada ao autor na área de textos do *applet*.

De forma a viabilizar as proposições da arquitetura do sistema, que prevê uma condição de interação direta do usuário com o AV, adversa das disponibilizadas pela GUI do protótipo, será necessária a definição de uma extensão para esta interface, uma vez que na proposição do sistema, é permitido ao autor selecionar um dado elemento no ambiente virtual, arrastá-lo para a posição que deseja e então definir seus parâmetros (interação direta).

A combinação apresentada na interface da Figura 5.15 é capaz de prover uma boa definição dos parâmetros e uma adequada realimentação das informações dadas pelo autor

ao *applet*, mas, não suporta o esquema de interação direta autor/meta-ambiente que atenda aos objetivos do sistema.

Isto posto, para verificar a adequação da continuidade de desenvolvimento com uso das linguagens adotadas nas definições dos protótipos, elaborou-se uma extensão da solução proposta inicialmente para a GUI, de forma a verificar a viabilidade da coleta de informações do cenário virtual pelo *applet*, que partilha a mesma página Web e é responsável pelo gerenciamento do processo de definição do AV desejado pelo autor.

Em tal interface, o desafio constituía-se em permitir que o autor selecionasse um dado objeto e o *applet* coletasse informações da seleção e da posição do cursor no AV. Se o protótipo desta interface provesse tal solução, problemas como a definição do elemento tocado pelo autor no AV e a posição definida pelo autor seriam sanados.

Com tal propósito, uma interface constituída de uma AV com dois carrinhos, combinado com um *applet* que recebe informações do AV foi desenvolvida e pode ser vista na Figura 5.20.
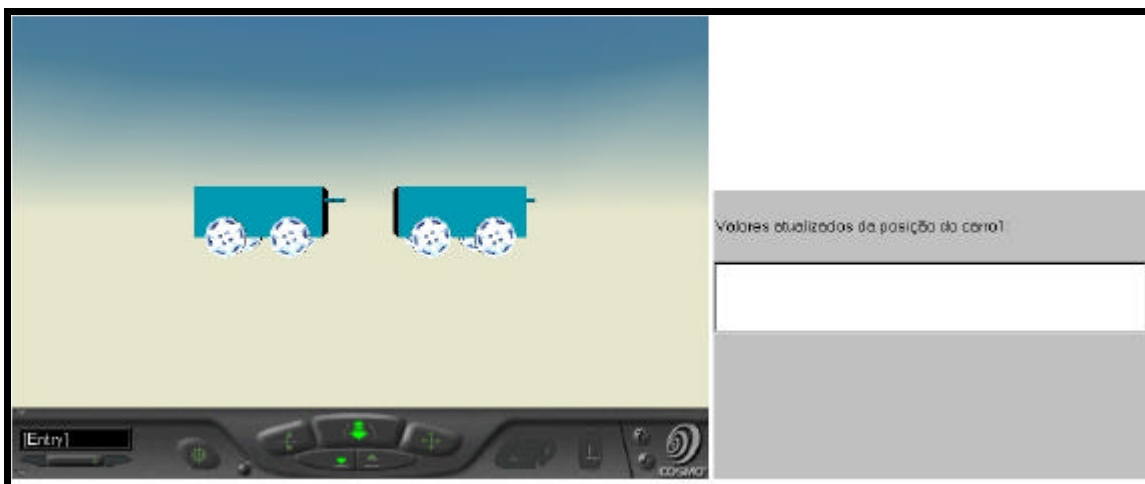
Fig. 5.20. - Protótipo de Interface de Coleta de Informações do AV

A Figura 5.21 apresenta o resultado coletado pelo *applet* Java, quando o autor seleciona o carro1 e arrasta-o para uma nova posição. O AV envia informações para o *applet* que as apresenta. O resultado deste desenvolvimento permite constatar a viabilidade de troca de informações que partam do AV para o *applet* Java, levando à conclusão que será simples adaptar a versão proposta na interface da Figura 5.15 para prover a interação

direta com o AV, de forma que seja possível armazenar as informações dadas pelo autor e salvá-las no arquivo que contém o AV desejado para um experimento.
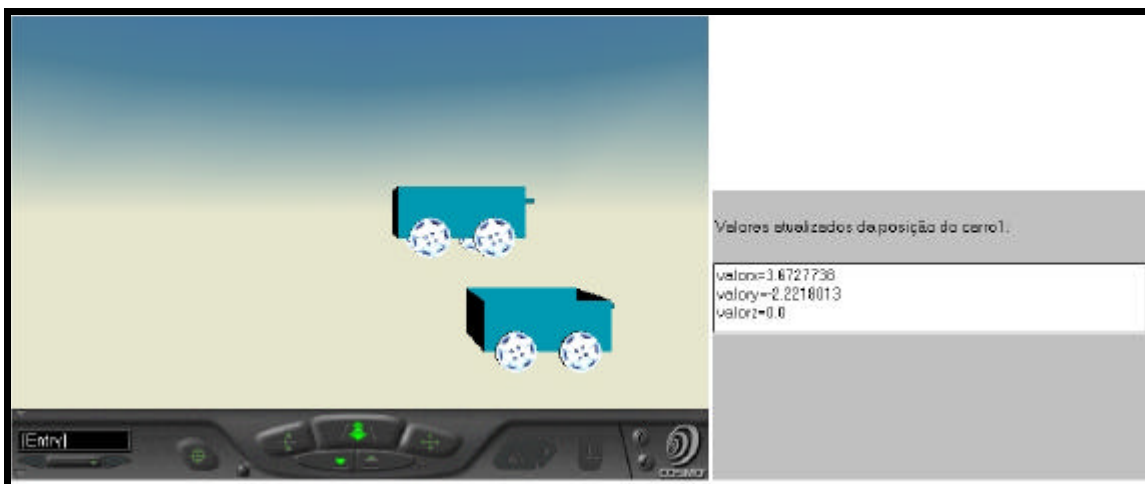
Fig. 5.21 - Resultados coletados pelo *applet* Java com alteração no AV

Para o retorno das informações do cenário virtual, o *applet* precisa capturar o nó e alguns valores que contêm as informações importantes. A Figura 5.22 apresenta o fragmento de código do *applet* que provê tal condição. A Figura 5.23 apresenta o fragmento de código que atualiza o texto relativo à posição do carrinho apresentado na Figura 5.21.

```
public void init() {
  browser = Browser.getBrowser(this);
  Node sensor = browser.getNode("TOUCH");
  Node sensor2 = browser.getNode("TOUCH2");
   isOver = (EventOutSFBool) sensor.getEventOut("isOver");
  isOver.advise(this, isOver);
        hitPoint_changed = (EventOutSFVec3f) sensor.getEventOut("hitPoint_changed");
        hitPoint_changed.advise(this, hitPoint_changed);
        isOver.advise(this, isOver);
        translation_changed = (EventOutSFVec3f) sensor2.getEventOut("translation_changed");
        translation_changed.advise(this, translation_changed);
        setLayout(new GridLayout(4,2));
        add(label);
        add(output);
 }
```

Fig. 5.22 - Código para Coleta de Informações de Nós VRML

```
public void callback(EventOut who, double when, Object which) {
        EventOutSFVec3f state = (EventOutSFVec3f) which;
                float f[] = new float[3];
                f = state.getValue();
                String temp = Float.toString(f[0]);
                String temp1 = Float.toString(f[1]);
                String temp2 = Float.toString(f[2]);
                output.setText("");
                output.appendText("valorx=");
                output.appendText(temp);
                output.appendText("\r");
                output.appendText("valory=");
                output.appendText(temp1);
                output.appendText("\r");
                output.appendText("valorz=");
                output.appendText(temp2);
                temp_who = String.valueOf(who);
                output.appendText(temp_who);*/
        }
       }
```

Fig. 5.23 - Código para Atualização das Informações do Cenário Virtual no *applet*

Uma das limitações desta forma de solução, aliando Java a VRML é relativa aos navegadores Web: algumas vezes tais navegadores não são compatíveis com as máquinas virtuais Java -JVM- ou com o *plug-in* VRML.

5.3.3. Subsistema de Experimentação

Visando dar condições à visualização e interação dos experimentadores com os experimentos virtuais, o subsistema de experimentação apresenta uma interface composta por um ambiente virtual que suporta:

- interações diretas com o cenário virtual do experimento;
- interações indiretas com um painel de controle.

Como exemplo de uma interface típica, que apresenta conceitos relativos ao sistema Massa-Mola, capaz de realizar um Movimentos Harmônicos Simples - MHS, o experimentador visualiza a utilização de um objeto mola, associado a um objeto massa. Neste caso, pode-se experimentar a modificação da constante elástica da mola, que é importante na análise de variáveis físicas do movimento, como o período de oscilação que é dado pela equação (1), onde m representa a massa do sistema fixo à mola e k a constante elástica da mola: $T = 2\pi \cdot \sqrt{\frac{m}{k}} \quad (1)$.

Para prover a modificação da constante elástica da mola e a visualização da oscilação, o autor definiu:

- um cenário virtual composto por uma mola e por uma massa (associadas);
- um painel de controle onde estão situados um atuador de disparo para visualização do movimento, um atuador que modifica a constante elástica da mola e um sensor que apresenta o valor atual da constante elástica.

Com tal configuração, uma mensagem enviada pelo objeto atuador altera a constante elástica do objeto mola e muda o comportamento do sistema, em termos de seu período de oscilação. A Figura 5.24 apresenta o resultado da definição visto na interface de experimentação.
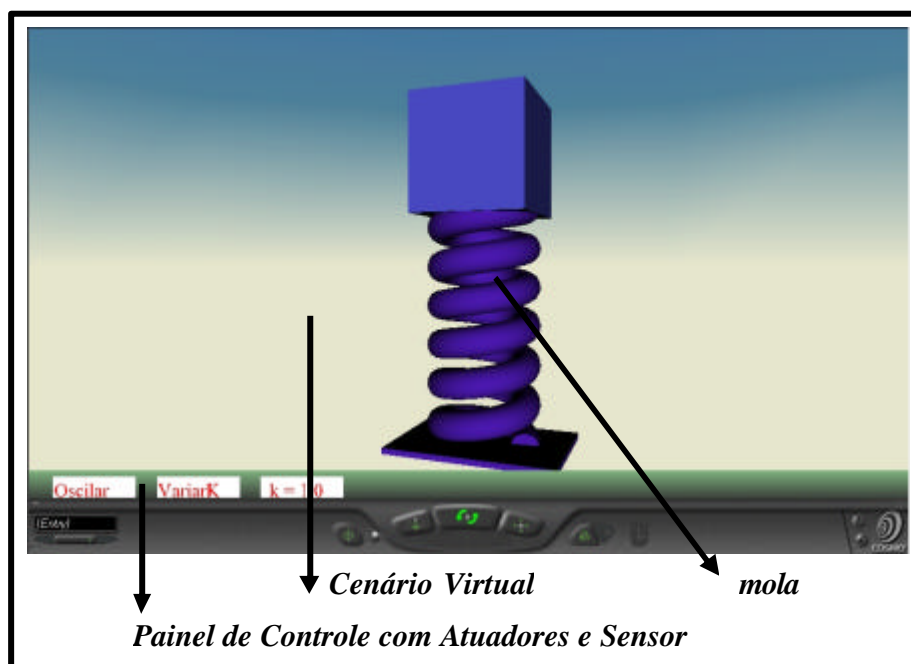
Fig. 5.24 - Sistema Massa-Mola

O painel de controle contém dois atuadores para controle: Oscilar e VariarK. Ao clicar sobre o botão Oscilar (atuador de ativação do experimento), a mola inicia um processo de oscilação. O botão VariarK (atuador de variação do valor da constante elástica) altera, dentre dois valores definidos, o valor da constante elástica da mola e o Sensor do painel de controle apresenta o valor corrente.

O sistema representado possui comportamento dinâmico, e permite a visualização da oscilação e da conversão de Energias. Considerando a troca de eventos entre objetos do cenário, o objeto Mola atualiza, por mensagem, a posição do objeto Massa, dando ao experimentador a noção do comportamento real do sistema. Em cada oscilação, uma nova espinha é gerada para a mola, de forma que a distância entre as espiras seja alterada.

De forma a apresentar as interações entre os objetos da interface, além das mensagens, são apresentados diagramas de seqüência relativos ao sistema Massa-Mola (Figura 5.25).
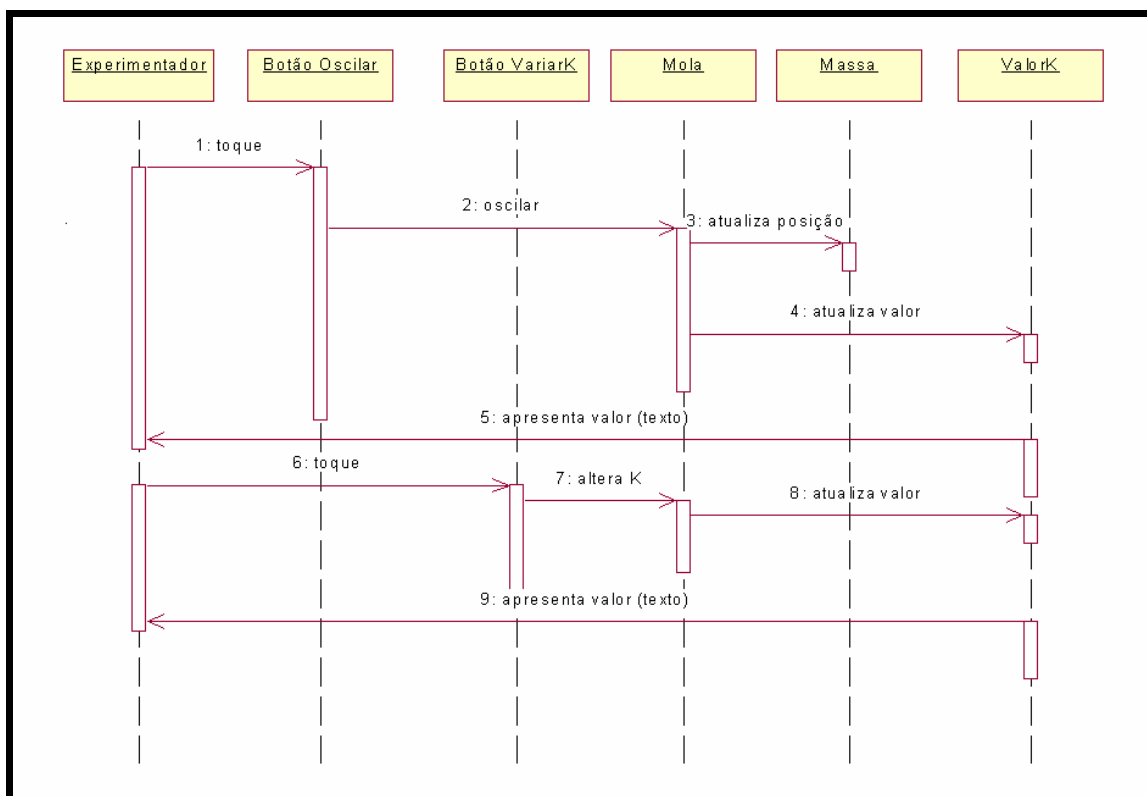
Fig. 5.25 - Diagrama de Seqüência de uma Oscilação

A implementação dos diversos objetos e das trocas de mensagens que são apresentadas na Figura 5.25 é efetuada com a combinação das linguagens VRML e JavaScript. A atualização de objetos do cenário virtual, tal como a deformação da mola e a posição da massa são implementadas por funções JavaScript inseridas no código fonte do cenário virtual, escrito em VRML.

A Figura 5.26 apresenta o código da função JavaScript que atualiza a deformação da mola e a posição da massa simultaneamente.

```
DEF Mudamola Script{
        field SFFloat spineRadius 1.5
        field SFInt32 spinePosition 0
        field SFInt32 spineResolution 32
        field SFFloat sepmola 0.04
        eventOut MFVec3f spine_changed
        eventIn SFFloat set_sepmola
        eventIn MFVec2f set_spineRadius
        eventIn SFInt32 set_spinePosition
        eventOut SFVec3f valor_pos
        url "javascript:
                function initialize( ){
                        generateSpine( )
                        }
                function set_spineRadius(sr,ts){
                        spineRadius = sr;
                        generateSpine( );
                        }
                function set_spinePosition(sp,ts){
                        spinePosition = sp;
                        generateSpine( );
                        }
                function set_sepmola(sm, ts){
                        sepmola = sm;
                        generateSpine( );
                        }
                function generateSpine( ){
                        y = spinePosition + 4;
                        angle = 0.0;
                        delta = 6.28 / spineResolution;
                        for( i = 0; i <= (spineResolution*6); i++){
                                spine_changed[i][0] = spineRadius * Math.cos(angle);
                                spine_changed[i][1] = y;
                                spine_changed[i][2] = -spineRadius * Math.sin(angle);
                                angle += delta;
                                y = y + sepmola;
                                        }
                        valor_pos[0] = 0.0;
                        valor_pos[1] = y + 1.8;
                        valor_pos[2] = 0.0;
                        }"      },
```

Fig. 5.26 - Código fonte que Atualiza a Deformação da Mola e a posição da Massa

O código retratado na Figura 5.26 demonstra a integração da linguagem JavaScript com VRML. As mudanças na espinha da mola providas pela função *generateSpine* atualizam o desenho da mola. Ao mesmo tempo, os valores da posição da massa (ou de qualquer outro corpo ligado a uma das extremidades da mola) podem ser definidos pelo vetor *valor_pos*. O código retratado é parte do código do objeto Mola definido na oscilação e presente no diagrama de seqüência da Figura 5.25.

O próximo capítulo discute a utilização dos protótipos e um estudo de caso relativo ao desenvolvimento de um experimento virtual relacionado com a Cinemática por um professor.

6. Resultados de Utilização

Este capítulo apresenta um estudo de caso relacionado à utilização do sistema para a criação de um dado experimento de Física, com utilização dos protótipos desenvolvidos e os resultados de avaliações dos protótipos por autores e experimentadores. Os protótipos desenvolvidos foram disponibilizados na Internet (Cardoso, 1998) e as avaliações foram efetuadas por meio de reuniões com avaliadores.

6.1. Estudo de Caso

Promovidas as implementações destacadas no capítulo anterior, protótipos do sistema foram criados, de forma a viabilizar que um dado autor (professor de Física) pudesse experimentar a elaboração de um experimento virtual relacionado com um dado tópico da Física.

O tópico escolhido está relacionado com o Princípio Fundamental da Dinâmica: a *Segunda Lei de Newton*, expressa em termos do resultado da aplicação de uma força resultante de módulo F sobre um corpo de massa m. A expressão vetorial da *Segunda Lei de Newton* é dada pela equação (1):

$$\vec{F} = m.\vec{a} \quad (1)$$

De maneira a introduzir tal conceito e possibilitar aos alunos experimentarem a relação entre a Força Resultante e a aceleração, Amaldi (1995), Alvarenga; Máximo (1998) propõem experiências envolvendo a comparação da aplicação de forças de mesmo módulo sobre dois ou mais elementos de massas diferentes, como carrinhos de laboratório. O resultado percebido pelo aluno, neste caso, é a variação da velocidade adquirida por corpos de massa diferentes sob ação de forças resultantes de mesmo módulo.

Em tais experimentos, os autores sugerem que alunos sejam levados a observar a velocidade que cada corpo adquire, que é função da aceleração a ele aplicada, através da experimentação. O resultado deve reforçar o conceito que a aceleração depende

inversamente do valor da massa do corpo, supondo a mesma força aplicada a todos os elementos envolvidos com o experimento.

Em condições reais de experimentação, fatores como o atrito e a resistência do ar acabam por interferir nos resultados mensurados pelos alunos. No entanto, o resultado final, caso o aluno promova o experimento, está associado à boa fixação do conceito fundamental da Dinâmica e a melhoria do aprendizado da noção de inércia de um corpo (relacionado com a *Primeira Lei de Newton*).

Visando introduzir tais experimentos, os protótipos do laboratório virtual desenvolvidos e disponibilizados aos autores apresentam:

(a) objetos de experimento relacionados a carrinhos que têm massa variável e à qual podem ser associadas outras massas;

(b) atuadores que permitem simular a aplicação de forças externas de diferentes intensidades nos objetos de experimentos;

(c) sensores relacionados a apresentar valores de massa, força e velocidade dos objetos de experimento e que sofrerão ação das forças externas;

(d) objetos de adorno relativos a simulação de superfícies de sustentação dos objetos de experimento, como mesas ou solos por onde os carrinhos serão arrastados.

Para criar um experimento a partir do laboratório virtual protótipo, o autor deve acessar o sistema, através da Internet, navegar no ambiente virtual do laboratório e de forma a criar um experimento relativo à Dinâmica, escolher o kit de desenvolvimento relacionado com a *Segunda Lei de Newton* dentre as diversas opções disponibilizadas. A Figura 6.1 apresenta uma parte do ambiente virtual, visualizada quando o autor opta por acesso aos kits de Dinâmica (simulação de um armário que contém alguns kits de laboratório).

Escolhido o kit, cabe ao autor a definição dos elementos que comporão o experimento virtual. No protótipo desenvolvido, tal escolha é feita através dos objetos de interface definidos em um *applet* Java que se comunica com o cenário virtual e com o painel de controle e permite a alteração dinâmica destas duas seções do Ambiente Virtual (AV). Conforme definido na especificação do sistema, as alterações providas pelo autor são imediatamente apresentadas no AV.
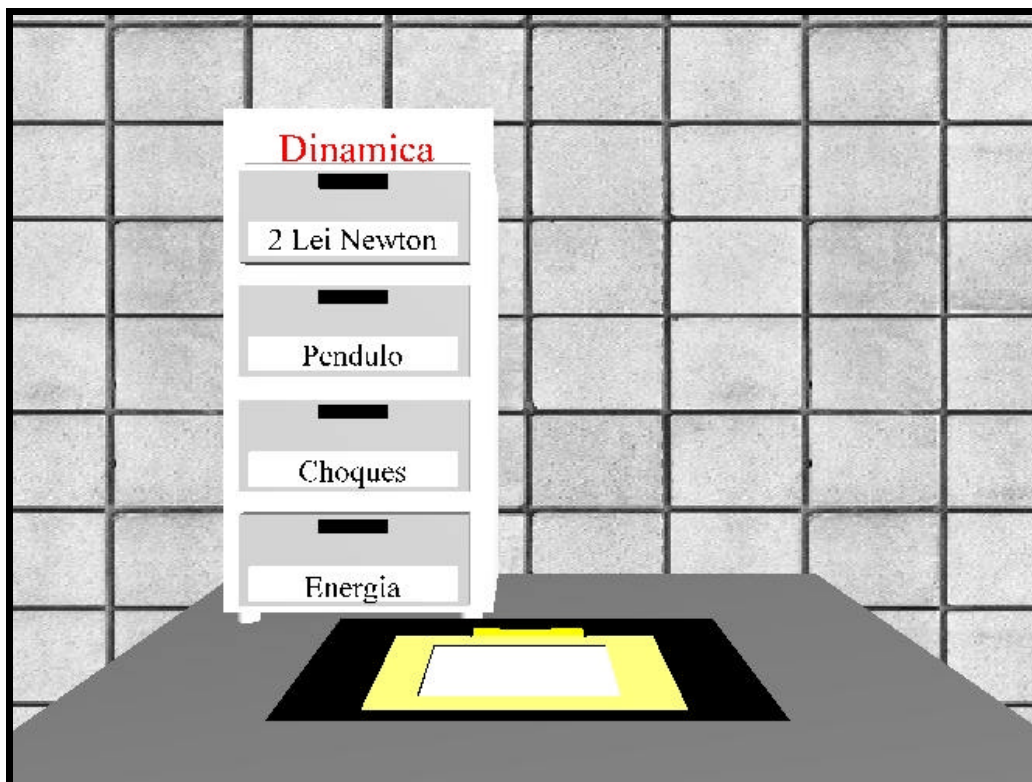
Fig. 6.1 - Selecionando o kit de Experimento Desejado

No estudo de caso relatado nesta seção, o autor optou por inserir três carrinhos de massas diferentes, conforme dados da Tabela 6.1.

Tabela 6.1 - Objetos de Experimento da Simulação

Objeto de Experimento	Massa do Objeto
Carrinho 1	*m = 1 um*
Carrinho 2 (com massa extra)	*m = 2 um*
Carrinho 3 (com massa extra)	*m = 3 um*

Supondo que os objetos sofram a ação de uma mesma força externa de módulo F, cada um deles apresentará diferente aceleração, velocidade final e espaço percorrido (supondo um dado intervalo de tempo), de forma que os experimentadores poderão visualizar movimentos variados para cada um dos objetos da simulação.

6.1.1. Definição de Objetos do Ambiente Virtual

De maneira a definir tal experimento virtual, o autor deve promover a inserção dos objetos no cenário e definir os componentes do painel de controle, que permitirão o disparo do processo de aplicação da força *F* sobre os carrinhos e a coleta de dados relativos à velocidade ou aceleração adquiridas pelos mesmos. Durante as inserções de objetos, de sensores e de atuadores no ambiente virtual, o autor deve promover:

1. a definição de atributos dos objetos, de acordo com suas pretensões, levando em consideração a natureza dos objetos e suas finalidades;
2. a definição dos relacionamentos, para definir as trocas de mensagens que são permitidas entre objetos sensores, atuadores e de cenário. Neste caso, cabe ao protótipo do sistema verificar associações inadequadas e não permitir definições incorretas;
3. a definição dos objetos de interface 3D que serão utilizados para atualização de atributos dos objetos do cenário virtual. Nesta hipótese, o autor pode escolher inserir um menu que modifique a massa ou a força aplicada a um dado carrinho à seleção do experimentador, dando condições de melhorias da interação com o experimento definido.

Neste estudo de caso, o autor optou pela seguinte seqüência de definição do experimento virtual:

1. definição do fundo de cenário e do tipo de superfície por onde os móveis se deslocarão, com uso da opção no *applet*, relativa à Figura 6.2;
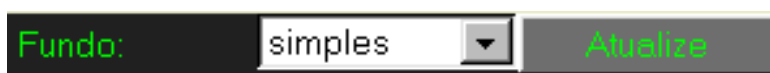
Fig. 6.2 - Menu de Seleção do Fundo de Cenário

2. inserção do primeiro objeto de experimento, denominado Carro 1, com a definição de atributos relacionados a este objeto, que são obrigatórios: valores de massa, força aplicada e posição do móvel no cenário virtual - Figura 6.3;

Fig. 6.3 - Definição de Atributos do *Carrinho 1* no Applet

3. inserção do segundo móvel e definição dos atributos, de forma que o segundo móvel tenha massa diferente do primeiro. O cenário virtual agora fica constituído de dois móveis diferentes. A Figura 6.4 apresenta uma tela relativa ao instante de definição da posição do segundo móvel, com destaque para o valor de *z*;
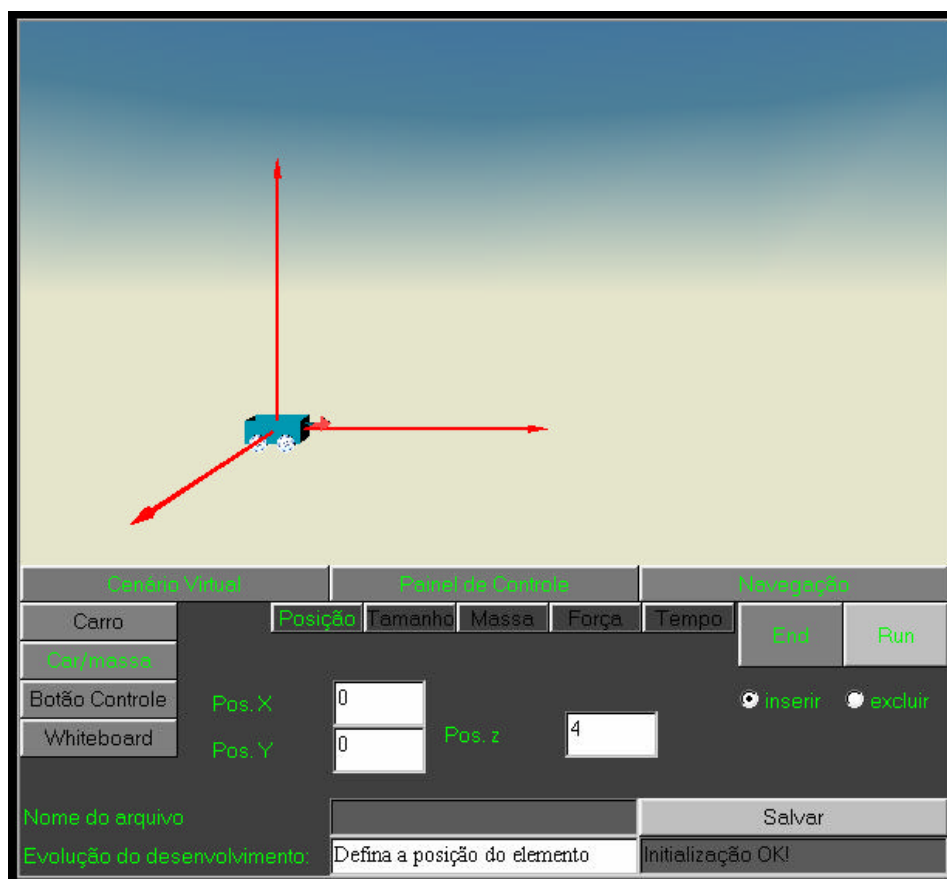
Fig. 6.4 - Definição de Atributos do *Carrinho 2*

4. definição do terceiro móvel, de forma que o cenário virtual fica constituído por três carrinhos de massas diferentes. Findo tal processo, o autor passou à definição do painel de controle e das relações entre os objetos atuadores e sensores, começando por um atuador (Figura 6.5) que proverá a simulação da aplicação da força sobre todos os móveis
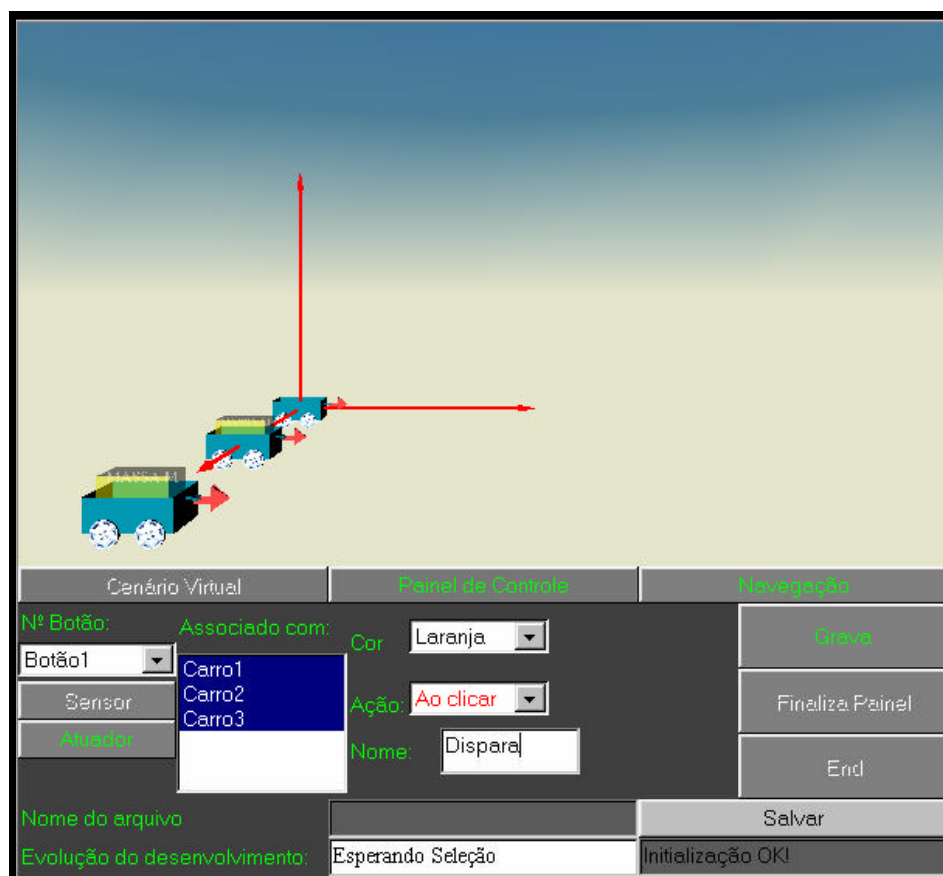
Fig. 6.5 - Definição de um Atuador para os três carrinhos

5. definição de sensores de velocidade e de espaço percorrido por cada móvel. A Figura 6.6 representa o instante da definição do sensor de velocidade do *Carrinho 2* relativo ao "Botão5". O autor deve definir o elemento a qual o botão será associado pela escolha do mesmo na lista de elementos do Cenário Virtual e a cor relativa ao botão que será inserido no Painel de Controle.

Fig. 6.6 - Definição de um Sensor para o *Carrinho 2*

Após todas as definições relativas ao Cenário Virtual e ao Painel de Controle terem sido efetuadas, o autor salvou o experimento (com a adequada atribuição de nome para o mesmo). A Figura 6.7 apresenta o Ambiente Virtual definido neste procedimento.
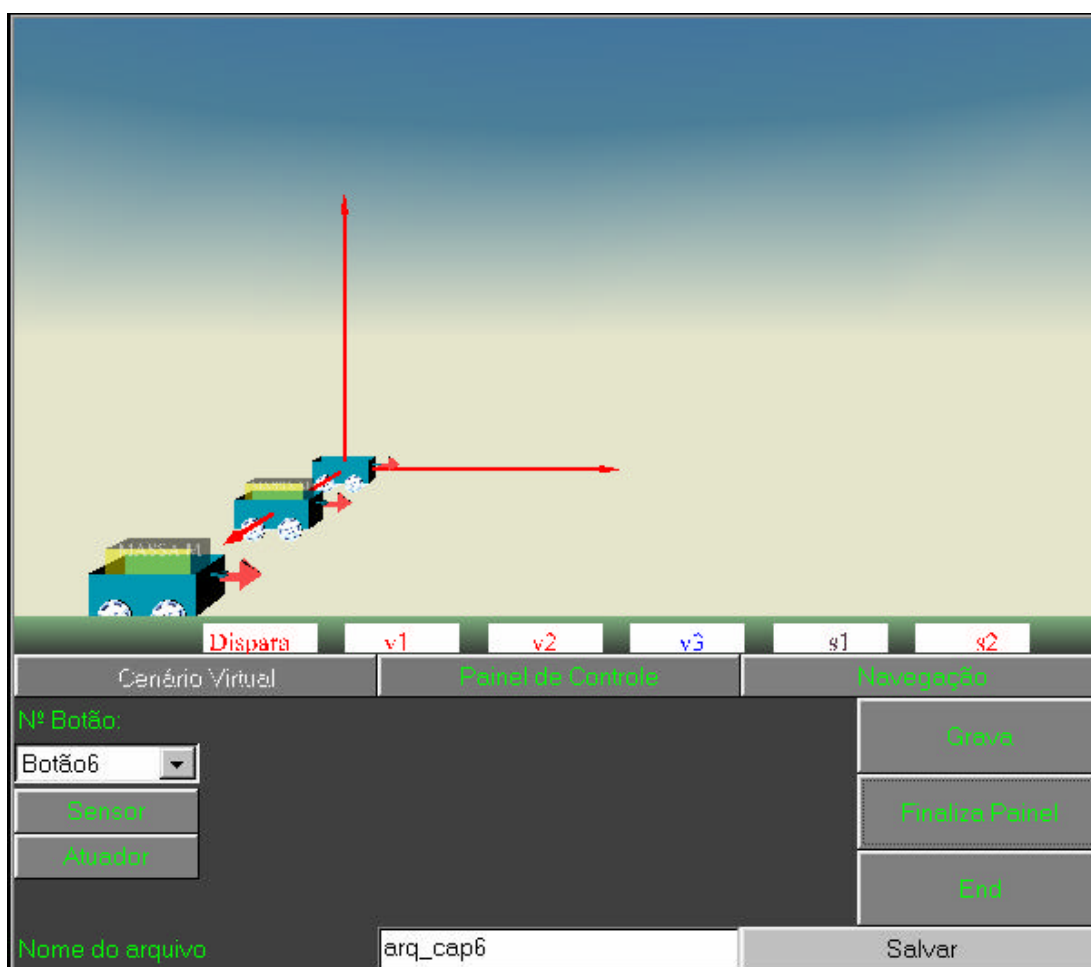
Fig. 6.7 - Cenário Virtual, Painel de Controle e Nome do Arquivo definidos

O arquivo relativo a este experimento virtual é armazenado no servidor e poderá ser carregado para uso por experimentadores através da Internet.

6.1.2. Recursos Adicionais à montagem do Experimento

Com respeito ao protótipo apresentado, algumas opções à simples montagem do experimento foram disponibilizadas. O autor, ao prover a definição dos componentes do cenário virtual, pode inserir quadros compostos de textos, com dicas, instruções ou fundamentos teóricos relativos ao experimento definido. O sistema formata o texto digitado pelo autor e insere o mesmo no Cenário Virtual, segundo os parâmetros fornecidos pelo autor.

Neste estudo de caso, o autor optou por inserir um quadro explicativo relativo à *Segunda Lei de Newton*, que é ativado na carga do experimento e é escondido através de um clique sobre o quadro, que pode ser visto na Figura 6.8.
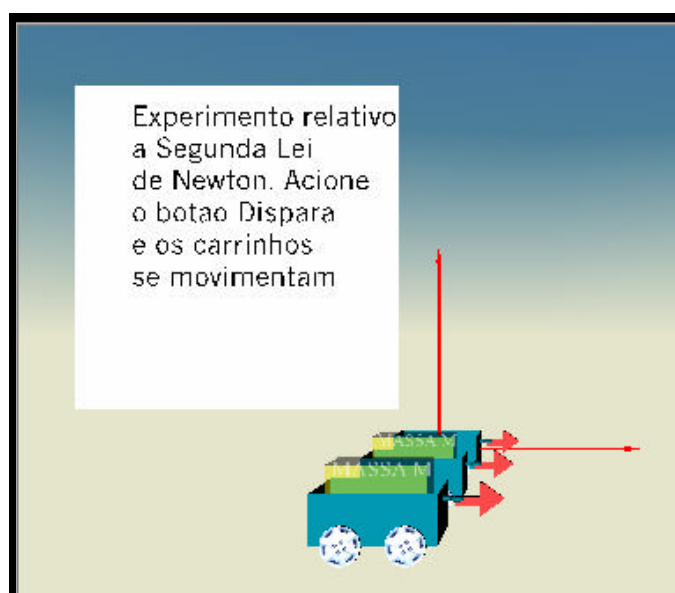
Fig. 6.8 - Quadro de texto inserido no Cenário Virtual

No processo de desenvolvimento, o autor definiu uma faixa de variação da massa e da força aplicada a cada carrinho do experimento. Esta opção visa possibilitar ao experimentador modificar os atributos dos componentes do Cenário Virtual durante a experimentação.

Pretende-se que o experimentador, como faria em um laboratório, possa alterar os seguintes parâmetros: (a) o valor da massa e (b) a força aplicada em cada carrinho durante a realização do experimento virtual. As faixas de variação seguem os valores definidos pelo autor, durante a criação do experimento. Para a atualização de valores, quando o experimentador clica sobre o carrinho inserido no Cenário Virtual, um *widget* 3D (Figura 6.9) é apresentado ao mesmo, que, por interação direta, poderá alterar os valores da massa e da força aplicada ao mesmo durante a realização dos experimentos.

Fig. 6.9 - *Widget* 3D para manipulação de parâmetros

Esta *widget* poderá ser manipulada pelo experimentador, suportando alteração de tamanho ou mudança de posição. Os botões permitem que seja incrementados ou decrementados, por meio de variações discretas dentro da faixa previamente definida, os valores dos parâmetros de experimento dos objetos, ou que a mesma seja destruída.

6.1.3. Utilização do Ambiente Virtual desenvolvido

Acessando o experimento virtual através da Internet, o experimentador obtém o ambiente virtual que permite a simulação relativa a este estudo de caso. A Figura 6.10 refere-se à página inicial obtida pelo acesso.

Pode-se observar o painel de controle, que contém o atuador, denominado "Dispara", para simular a aplicação das forças nos móveis e permitira a realização do experimento, os sensores de velocidade dos móveis 1, 2, 3 (v1, v2, v3, respectivamente) e de espaço percorrido pelos móveis 1 e 2 (s1, s2, respectivamente).
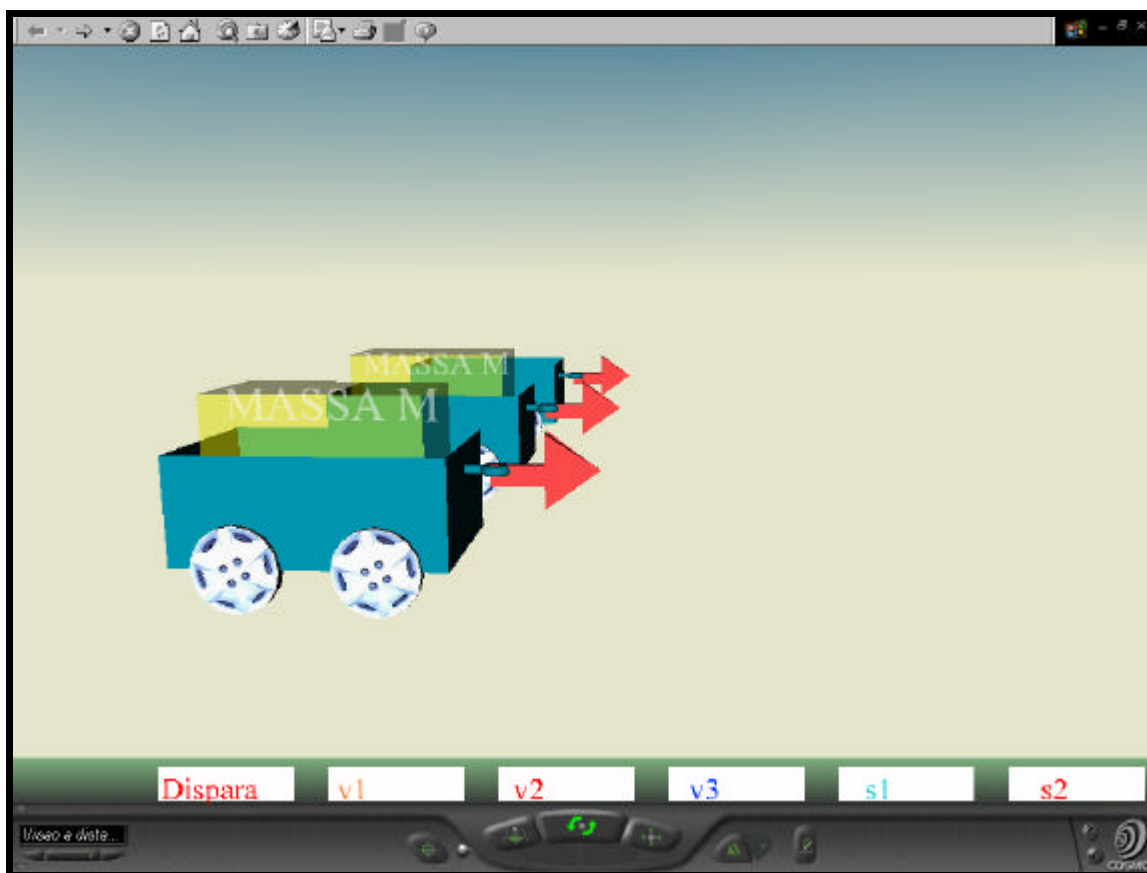
Fig. 6.10 - Página Web inicial do experimento da Segunda Lei de Newton

Com o clique do experimentador sobre o atuador "Dispara", os três carrinhos são submetidos à experiência. A Figura 6.11 apresenta as posições dos móveis após a aplicação da Força e a Figura 6.12 apresenta detalhes dos sensores de velocidade associados ao painel de controle.

O experimentador, por interação direta com os elementos do Cenário Virtual pode atualizar valores dos atributos dos objetos, visando uma nova experimentação, através dos menus 3D disponíveis para cada móvel. Os valores de faixa de atributos estabelecidos na definição do experimento é que estão disponíveis em cada menu.
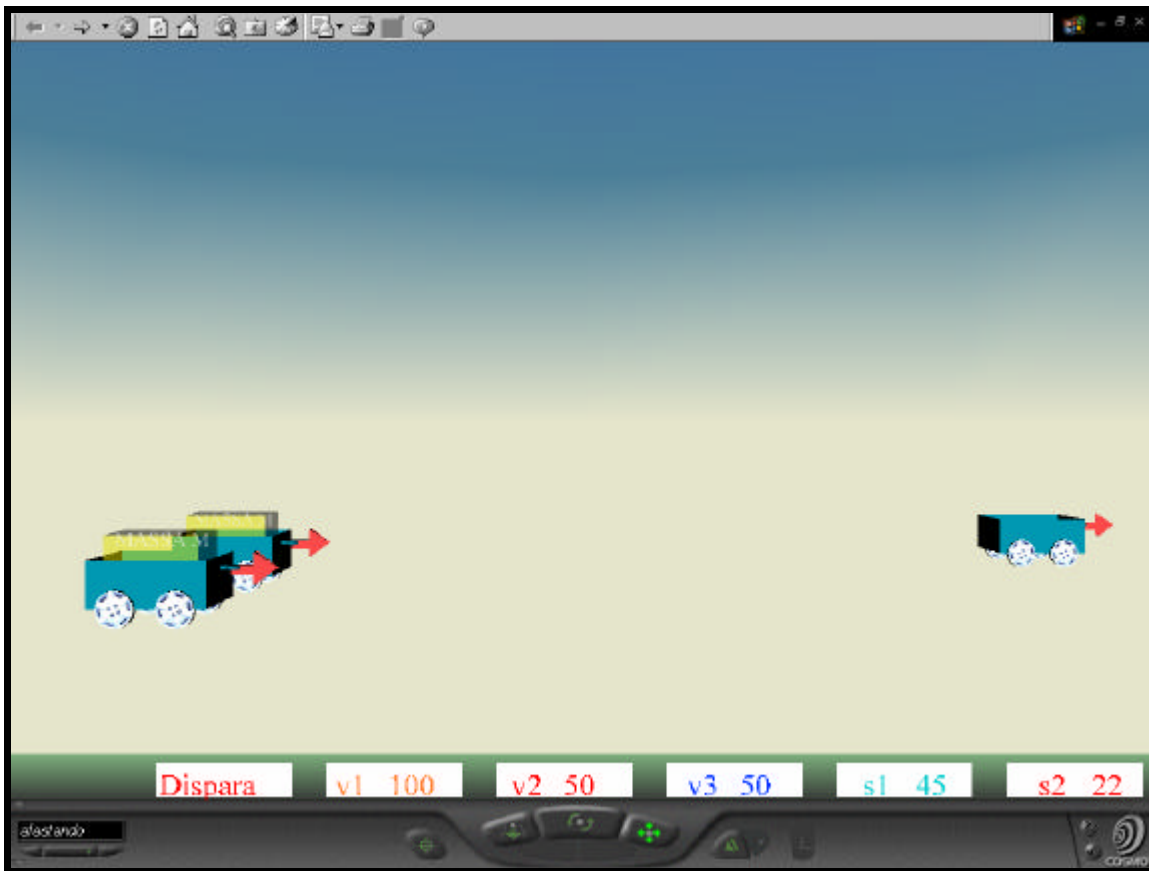
Fig. 6.11 - Cenário Virtual e Painel atualizados pela movimentação dos móveis
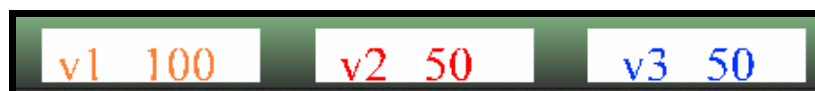
Fig. 6.12 - Valores indicados nos Sensores de Velocidade (Painel de Controle)

No estudo de caso apresentado, o autor definiu que o *Carrinho 2* podia ter massa variando de 2 a 8 u.m. e Força Resultante variando de 1 a 5 N. Com a modificação dos valores da Força aplicada a este móvel para 2N, teríamos sua movimentação similar à que ocorre para o *Carrinho 1*. A Figura 6.13 apresenta o resultado desta simulação considerando tais modificações.
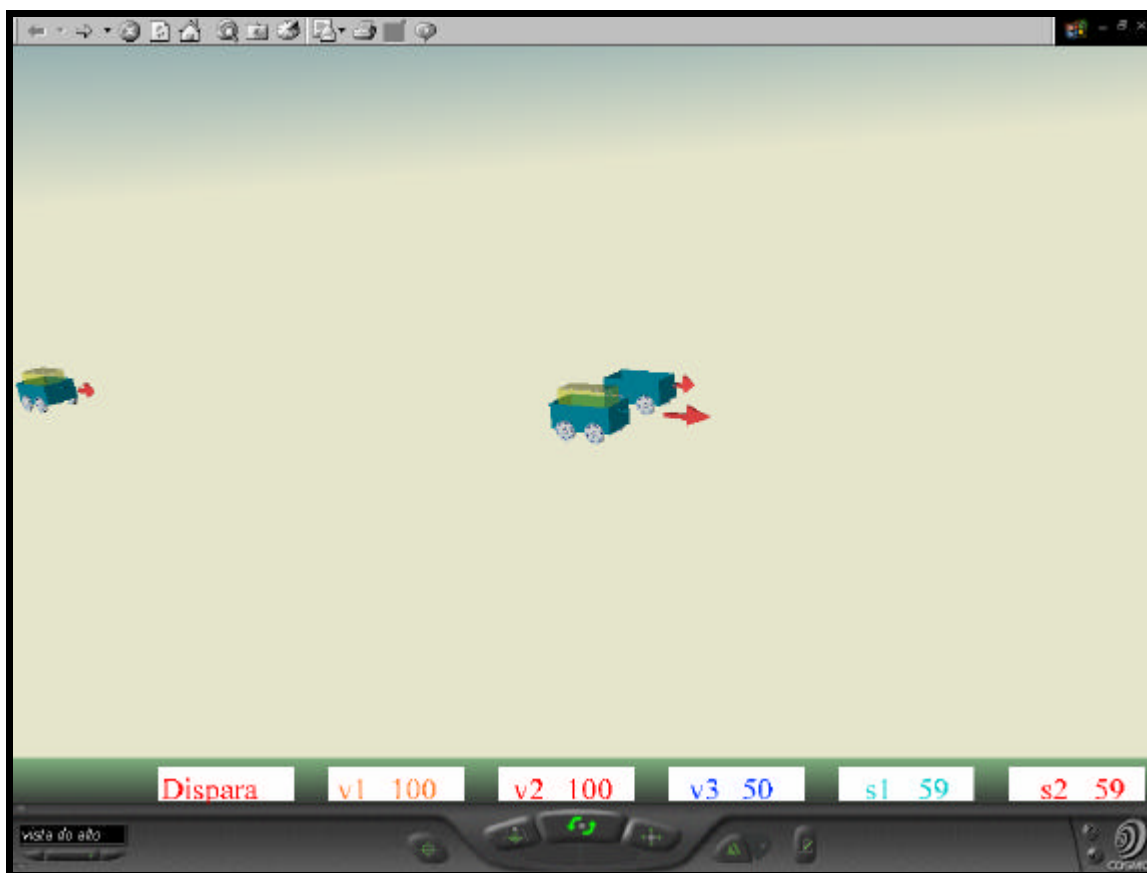
Fig. 6.13 - Resultado do experimento com alterações no *Carrinho 2*

6.2. Resultados de Avaliação dos Protótipos

Através de reuniões com professores de Física do segundo grau, de três escolas da cidade de Uberlândia, sendo duas delas particulares e uma delas da rede estadual de ensino, foram apresentados os protótipos:

1. dos cenários comuns do laboratório virtual;
2. de exemplos de experimentos desenvolvidos, como o sistema massa-mola, da imersão de uma espira condutora em Campo Magnético - Fluxo Magnético (Figura 6.14); lançamento de cargas elétricas em campos elétricos (Figura 6.15) e magnéticos.
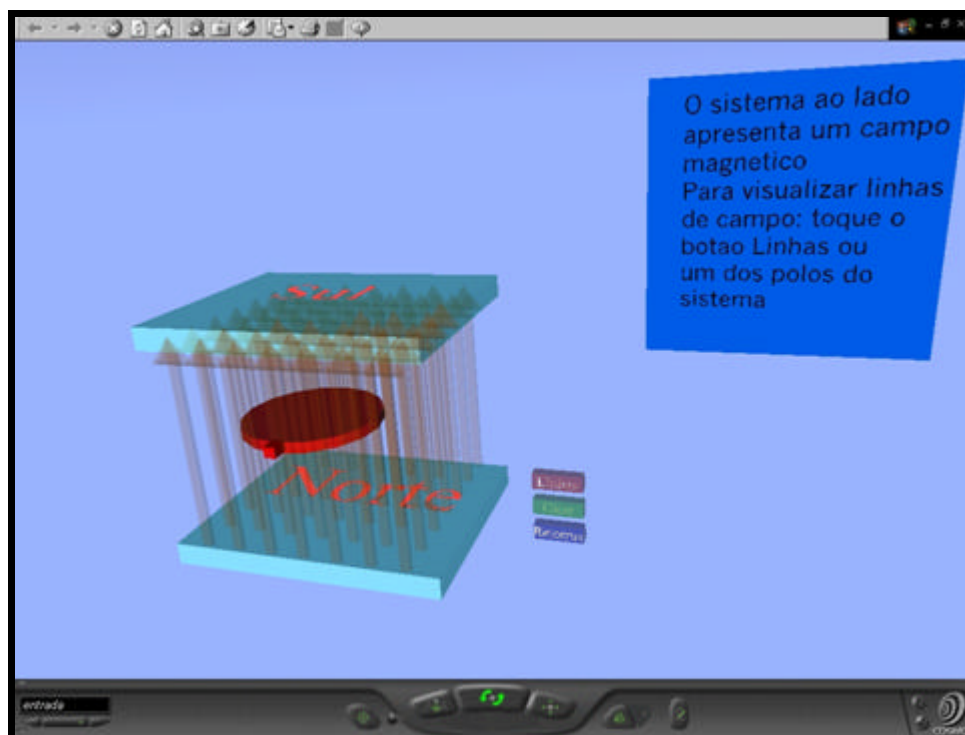
Fig. 6.14 - Experimento de Imersão de Espira em Campo Magnético
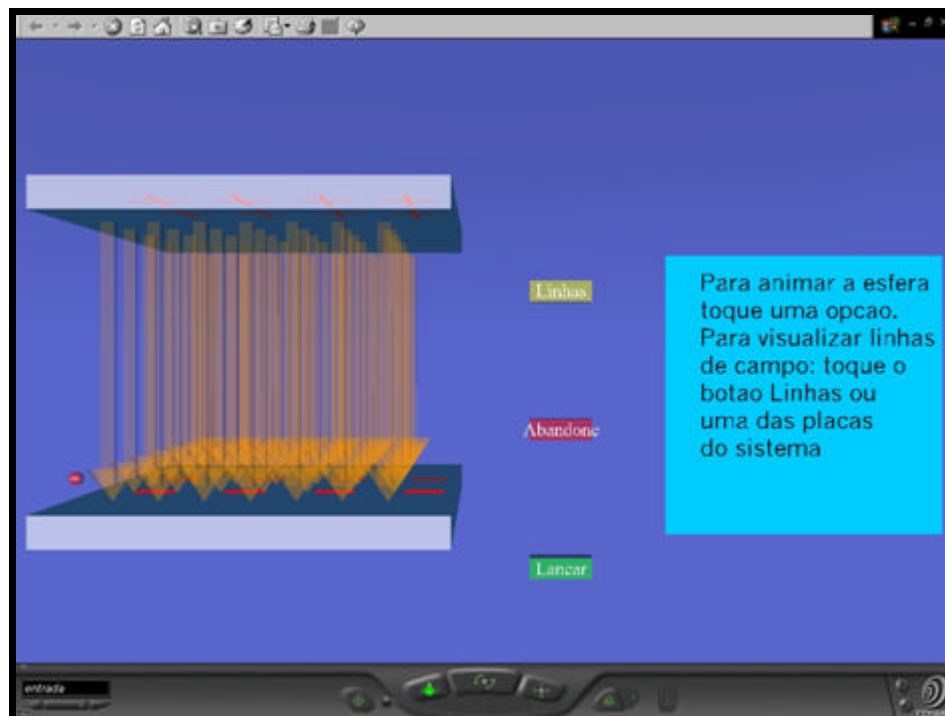
Fig. 6.15 - Experimento de Lançamento de uma partícula em Campo Elétrico

3. do protótipo que provê o desenvolvimento de experimentos virtuais relativos à *Segunda Lei de Newton*, relacionados com o estudo de caso do tópico anterior, permitindo que o avaliador pudesse verificar o desenvolvimento e utilizar um experimento virtual definido por ele mesmo.

Nos diversos encontros com os avaliadores, foram solicitados preenchimentos de questionários de avaliação dos protótipos, conforme o modelo apresentado no Anexo II.

Os resultados de avaliação dos usuários apresentam resultados promissores, uma vez que os professores apontam que o software proposto e avaliado é apontado como uma ferramenta importante para a melhoria do trabalho deles no ensino de Física e de grande finalidade. Ao mesmo tempo, os avaliadores indicaram que a interface desenvolvida nos protótipos é de fácil entendimento e que os mesmos são de fácil utilização.

Os itens listados abaixo apontam algumas sugestões dos avaliadores:

i) melhoria relativa a ampliação da biblioteca de objetos de adorno e de experimentos;

ii) aprimoramento do *help* com inserção de exemplos ilustrativos dos passos de composição dos ambientes virtuais;

iii) que o sistema deve apresente previamente todos os objetos disponíveis no kit de experimento, de forma que o usuário faça a seleção do objeto dentro do ambiente virtual e não através de uma lista;

iv) melhoria do sistema no sentido de prover mais kits de experimentos.

Da análise e do acompanhamento das avaliações, pode-se observar que há certa dificuldade de entendimento de alguns comandos, relativos ao *plug-in* VRML (neste caso, o CosmoPlayer), que parece simples aos mais familiarizados, mas, que algumas vezes não é intuitivo aos usuários

Além do mais, os arquivos de *help* dos protótipos não conseguiram apresentar respostas a todas as dúvidas dos usuários, que, às vezes, se perdiam em uma ação e não sabiam como retornar dela.

Uma das observações importantes dos avaliadores refere-se ao fato do sistema não apresentar todos os objetos disponíveis, ao mesmo tempo, e não exclusivamente os relativos ao *kit* em questão, permitindo associações entre objetos quaisquer.

A queixa relativa ao número de *kits* foi muito comum, o que leva à conclusão que o usuário gostaria de poder desenvolver outros experimentos virtuais, o que aponta uma grande motivação dos mesmos a utilizar o sistema para os diversos tópicos da Física.

6.3. Limitações dos Protótipos

A partir da escolha das linguagens e do desenvolvimento, algumas limitações dos protótipos puderam ser constatadas:

i) em alguns tipos de processadores, a velocidade de navegação pelos cenários virtuais é muito baixa, principalmente quando não há placa aceleradora 3D;

ii) a integração de Java e VRML através da *EAI* apresenta muitas limitações, no que diz respeito à captura de informações pelos *applet* Java a partir dos nós VRML. Estas limitações forçam o programador a usar uma série de estratégias para contorná-las e atingir os objetivos desejados;

iii) a opção de inserir texturas é útil na melhoraria da qualidade dos objetos de cenário, no entanto, a inserção de imagens aumenta muito o tamanho dos arquivos e piora a velocidade do processamento das animações, levando à conclusão que a inserção de texturas precisa ser bem avaliada, numa solução de compromisso entre aprimorar o realismo da simulação e de velocidade do tratamento da imagem.

É importante considerar que as limitações dos protótipos não são limitações da arquitetura. A evolução das linguagens utilizadas nos protótipos pode possibilitar que algumas das restrições atuais dos protótipos sejam acomodadas, dando melhores condições de desenvolvimento do projeto do sistema. Neste cenário, o constante aprimoramento de Java e a evolução de VRML para o X3D pode definir o investimento na elaboração de novos protótipos, ou até mesmo da solução completa da arquitetura, com utilização destas linguagens. Visando uma discussão mais elaborada de tais expectativas, o capítulo seguinte aborda as principais conclusões e os trabalhos futuros.

7. Conclusões e Trabalhos Futuros

7.1. Preâmbulo

Esta tese apresentou uma arquitetura para um meta-ambiente (provendo uma infra-estrutura para concepção de ambientes virtuais) relacionado a um dado domínio de aplicação. Alguns protótipos de partes desta arquitetura foram implementados, com utilização de linguagens de programação compatíveis com a Internet. Finalmente, o trabalho apresentou uma avaliação efetuada por professores do nível médio dos protótipos e da proposição da arquitetura.

7.2. Conclusões

A partir da revisão bibliográfica e do processo de análise e projeto do sistema, pôde-se constatar que:

i) há demanda por sistemas capazes de proporcionarem a usuários desprovidos de conhecimentos de técnicas de programação e Computação Gráfica a desenvolverem seus próprios ambientes virtuais - meta-ambientes, desobrigando os mesmos do aprendizado de técnicas específicas, tais como a associação de transformações etc;

ii) a definição das classes do sistema de forma *bottom-up*, ou seja, a partir da definição de objetos e seus atributos contribui para uma boa separação de classes (permeando um bom agrupamento por funcionalidade dos objetos) envolvidas com a solução do meta-ambiente que gerará os experimentos virtuais desejados;

iii) a utilização de módulos de compiladores (analisadores léxicos e sintáticos e geradores de código) é indispensável à arquitetura dos meta-ambientes, já que tais módulos executam papéis importantes na interpretação de comandos dos usuários, na verificação de erros e na geração dos códigos dos ambientes virtuais definidos pelos usuários do sistema;

iv) a arquitetura proposta e a implementação apresentam vantagens relativas à portabilidade e adequação a propósitos de educação presencial e a distância, relacionadas com a utilização de linguagens de programação independentes de plataforma (VRML, Java, JavaScript) compatíveis com a Internet e RV não-imersiva.

Com a utilização dos protótipos, desenvolvidos com base nos aspectos definidos para a arquitetura do sistema, pôde-se verificar:

i) a utilização de linguagens de programação independentes de plataforma (como Java, VRML e JavaScript) e a posterior produção de material compatível com a Internet provê facilidades em relação a soluções que impõem a necessidade de programas auxiliares (como a proposta do sistema Alice). Destacam-se como facilidades: a dispensa de necessidade de baixar programas específicos para utilização de um dado sistema (e instalá-los); a possibilidade de adequação de versões sem a necessidade de exigir do usuário constantes atualizações e a preservação da independência de plataformas;

ii) a união de VRML com JavaScript é indispensável para a descrição comportamental dos objetos. JavaScript é utilizada para a construção das operações dos objetos da arquitetura. A opção por inserir o código JavaScript dentro dos objetos VRML foi fundamental para permitir a descrição comportamental dos mesmos e viabilizar os protótipos;

iii) na associação de VRML com Java, através da EAI, ainda há diversos problemas, principalmente relativos a retorno da informação gerada pelos componentes do cenário virtual apresentado por um *plug*-in VRML. Neste aspecto, mesmo com a troca do *plug*-in e das classes Java associadas, a implementação enfrentou problemas, tais como as limitações das informações derivadas dos nós VRML para os *applets*, que necessitam estarem associadas a um nó VRML previamente definido;

iv) os diversos navegadores da Internet apresentam limitações ao uso combinado de VRML, Java e JavaScript. No caso do Netscape, a utilização de JavaScript (associado a HTML) gerando códigos em VRML é viável, já

que o navegador é capaz de atualizar um dado *frame* relativo ao código VRML, sem perda da informação. A mesma proposição não funciona adequadamente com o Internet Explorer (versão 5.5). Quanto à combinação Java, VRML e HTML, pôde-se constatar que erros nos *applet* Java geralmente não originam mensagens de erro e causam o travamento do programa navegador.

A partir das avaliações dos protótipos por professores, desprovidos de conhecimentos de técnicas de programação e de Computação Gráfica, pôde-se concluir:

i) a implementação da arquitetura é avaliada como muito útil, viabilizando a solução de um laboratório de conteúdo exclusivamente virtual que pode enriquecer e complementar o ensino de Física;

ii) a arquitetura proposta permite a realização (interativa) de experimentos de Física por alunos do Ensino Médio, podendo ser utilizada no ensino presencial ou a distância;

iii) os avaliadores ficam motivados com a possibilidade de utilização de técnicas de Computação Gráfica e RV sem a necessidade de prévio conhecimento de técnicas específicas ou do aprendizado de tecnologias;

iv) a solução proposta na concepção de experimentos virtuais permite a visualização, interação e experimentação de tópicos geralmente qualificados, pelos aprendizes, como "muito abstratos" ou de "difícil visualização", que relatam satisfação no ato de participar de uma dada experiência.

7.3. Trabalhos Futuros

Considerando a arquitetura proposta e o desenvolvimento dos protótipos deste trabalho, pôde-se visualizar os trabalhos futuros relativos à arquitetura e ao desenvolvimento de protótipos e à implementação do sistema proposto.

7.3.1. Extensões da Arquitetura do Sistema

Considerando os aspectos relacionados à arquitetura do sistema proposto, são identificados como trabalhos futuros:

i) a extensão da arquitetura do meta-ambiente proposto a outros domínios do conhecimento, tais como Química ou Matemática, bem como analisar a

viabilidade do uso das técnicas propostas no desenvolvimento de novas arquiteturas que suportem multidisciplinaridade, onde a implementação de laboratórios virtuais possa ser utilizada como auxílio no ensino;

ii) a definição de novos objetos de interface e de interação com os usuários, tais como os "menus 3D", de forma que o sistema possa ser implementado com uso de tecnologias de RV imersiva, capacitando a navegação, definição de sensores e atuadores e manuseio de objetos de adorno e de experimentos.

7.3.2. Extensões de Implementação

Considerando o desenvolvimento dos protótipos e os aspectos arquiteturais, são identificados como trabalhos futuros:

i) a implementação da interface e dos módulos relativos ao Subsistema de Atualização de Objetos, permitindo que um professor conhecedor de linguagens de programação ou um programador VRML (qualificado como "desenvolvedor" na definição do sistema) possa inserir novos objetos, por ele elaborado (compatíveis com as definições de classe) no sistema;

ii) o aprimoramento e a extensão da biblioteca de objetos, com aumento da quantidade de objetos relativos a objetos de adorno, de experimentos e de interface;

iii) o desenvolvimento de novos protótipos com utilização de outras linguagens de programação, em especial a verificação da adequação a este propósito da linguagem X3D, dita sucessora do VRML 2.0, que apresenta melhorias nos aspectos relacionados a orientação a objetos, através de nós *proto* que permitem melhor definição e reutilização de objetos;

iv) a evolução dos protótipos do sistema de forma a suportarem toda a definição do experimento virtual desejado por interação direta do autor com elementos do laboratório virtual, contemplando os requisitos da arquitetura. Desta forma, seriam geradas interfaces que permitem que um dado autor selecione (dentre uma série de objetos disponíveis no cenário virtual) um único objeto, arraste-o para uma posição particular (como ficar sobre uma mesa de laboratório) e defina os atributos dos objetos que comporão um dado experimento, como se estivesse em um laboratório real.

Anexo I

Diagrama de Classes do Sistema

Anexo II

Avaliação do Programa Gerador de Experiências de Física

Avaliador: ____Data da Avaliação: ___/____/20__

Habilitação: ________________________________ ( ) 1º grau ( ) 2º grau
Escolas onde leciona:
____tel( ) ____________

____tel( ) ____________

____tel( ) _____________

Avalie o Programa Gerador de Experiências de Física:

I. Quanto à Finalidade:

( ) Muito útil ( ) útil ( ) pouco útil

( ) Aplicável ao meu trabalho atual, sem exigir adequações do local de trabalho;

( ) Aplicável ao meu trabalho atual, exigindo poucas adequações do local de trabalho; (Quais??)

____
____
____
____

( ) Aplicável ao meu trabalho atual, exigindo muitas adequações do local de trabalho; (Quais??)

____
____
____
____
____

( ) Não aplicável ao meu trabalho atual. (Justificativa)

____
____
____
____
____

II. Quanto à Interface:

( ) Fácil entendimento dos comandos
( ) Médio entendimento dos comandos
( ) Difícil entendimento dos comandos

III. Quanto a facilidade de uso

( ) Muito intuitivo;
( ) Intuitivo;
( ) Pouco Intuitivo.

IV. Quanto à seqüência de operações

( ) A ação seguinte é clara
( ) A ação seguinte é obscura

V. Quanto aos recursos do Programa, a experiência proposta :

( ) foi integralmente desenvolvida.
( ) não foi desenvolvida por completo.

V.b. Os objetos disponíveis permitem :

( ) conceber a experiência proposta
( ) conceber parte da experiência proposta
( ) não permitem conceber a experiência.

V.c. A relação entre os objetos disponíveis e suas propriedades:

( ) é simples de ser definida e permite o desenvolvimento proposto;
( ) não é simples de ser definida, mas permite o desenvolvimento proposto;
( ) não é simples de ser definida e não permite o desenvolvimento proposto;

VII. Sugestões Adicionais:

Assinatura do Avaliador: ______________________________

Lista de Referências

ALICE v. 1998, ALICE Software. Página da Internet do Software "Alice" que disponibiliza o software e o *plug-in* para utilização do mesmo na Internet. Carnegie Mellon University. Disponível em <http://www.alice.org>. Data de acesso: 20 de nov. 2000.

ALLISON, D. et al. The Virtual Gorilla Exhibit. **IEEE Computer Graphics and Applications**, USA, p. 30-38, Nov./Dec. 1997.

ALVARENGA, B., MÁXIMO A. Segunda Lei de Newton. In: **Curso de Física.** 4.ed, Editora Scipione, São Paulo, Brasil, 1998. p.229-295

AMALDI, U. **Imagens da Física - As Idéias e as Experiências do Pêndulo aos Quarks**, Tradução, Editora Scipione, São Paulo, Brasil, 1995

AMES, L. A.; NADEAU, R.D.; MORELAND D. **VRML Sourcebook - Second Edition**, John Wisley & Sons, Inc - USA, 1997.

ANDRADE, F. A.; WAZLAWICK, S. R.; MARIANI C. A. Metodologia para Criação de Roteiros Educativos em Realidade Virtual. **Revista Brasileira de Informática na Educação**, Brasil, p. 69-76, setembro 1999.

ARAGÃO, M.R.R.; RIVERO, L.M.C., BACCARIN, F. Educação a Distância: Em Busca de Critérios Didático-Pedagógicos nas Práticas Existente. In: XX CONGRESSO NACIONAL DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO - **SBC 2000**, vol.1, Curitiba - PR, 2000, p.85

ARETIO, L.G., **Educación a distancia hoy**. Madrid: UNED (Colec. Educación permanente), 1994

BASAL, F. Virtual Reality and schools project report - VR in the Schools. Jornal online sobre aplicações de Realidade Virtual na Educação, com apresentação de experiências e resultados. Disponível em <http://eastnet.educ.ecu.edu/vr/vrlnla.txt>. Acesso em: 20 de jun. 1999.

BELL, J.; FOGLERL H. S. The Investigation and Application of Virtual Reality as an Educational Tool. In: PROCEEDINGS OF THE AMERICAN SOCIETY FOR ENGINEERING EDUCATION ANNUAL CONFERENCE, Anaheim, CA, June 1995.

BOOCH G.; RUMBAUGH J.; JACOBSON I., **UML - Guia do Usuário**, Editora Campus, Rio de Janeiro - Brasil, 2000.

BORK, L. A. ; BRITTON, R. D, The Web is Not Yet Suitable for Learning. **IEEE Transactions on Computer,** USA. p. 115-119, June 1998.

BOULET, M. M. Distance learning fo the management of software projects. In: ACM PROCEEDINGS OF THE CONFERENCE ON INTEGRATING TECHNOLOGY INTOCOMPUTER SCIENCE EDUCATION – **ITICSE´97**, 1997, p. 136-138.

BYRNE, C. **Water on Tap: The Use of Virtual Reality as an Educational Tool**. Washington, 1996. Tese (Doutorado) - University of Washington.

BRYSON, S. et al. The Future of Virtual Reality: Head Mounted Versus Spatially Immersive Displays. In: PROCEEDINGS OF THE SIGGRAPH'96 CONFERENCE ON COMPUTER GRAPHICS, 1996, p. 485-486.

CAMPOS, C. F. **Novo Glossário de Informática.** Editora Ciência Moderna, Rio de Janeiro, Brasil, 1995.

CARDOSO, A.; LAMOUNIER E.; TORI R. Sistema de Criação de Experiências de Física em Realidade Virtual para Educação a Distância. In: II WORKSHOP BRASILEIRO DE REALIDADE VIRTUAL, **WRV´99**, Marília , São Paulo, Brasil, 1999, p. 174-181.

CARDOSO, A.; LAMOUNIER E.; TORI R. A Virtual Physics Laboratory to Support Interactive Distance Learning In: INTERNATIONAL CONFERENCE ON ENGINEERING AND COMPUTER EDUCATION - **ICECE´2000** - São Paulo, S.P., ago. 2000, 1 CDRom.

CARDOSO, A.; LAMOUNIER E.; TORI R. Realidade Virtual no Ensino de Física a Distância e Presencial In: XI SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO - **SBIE´2000,** Maceió, Alagoas, nov. 2000- p. 476-478.

CARDOSO A.; LAMOUNIER E.; TORI R. Interactive 3D Physics Experiments Through the Internet. In: 4TH SYMPOSIUM ON VIRTUAL REALITY- SVR 2001. Florianópolis, SC, Oct. 2001, p. 280-290.

CARDOSO, A. Alexandre Cardoso - Página do Pesquisador. Contém informações sobre aplicações de Realidade Virtual, pesquisa e publicações do pesquisador, tutoriais sobre VRML e artigos indicados para leitura. Disponível em: <http://www.compgraf.ufu.br/alexandre/alexandre.html>. Acesso em Jan. 1998

CARSWELL, L. Teaching via the Internet: The Impact of the Internet as a communication medium on Distance Learning Introductory Computing Students. **ACM Sigse Bulletin** v. 29, n. 3, p. 1-5, Sep. 1997.

CHAN, M.; GRIFFITH, W. S.; IASI, A. **Java 1001 Dicas De Programação.** 1ªEdição, Editora Makron Books, São Paulo, 1999.

CHEN, S.; MYERS, R.; PASSETO, R. The Out of Box Experience - Lessons Learned Creating Compelling VRML 2.0 Content. In: Proceedings of the Second Symposium on Virtual Reality Modeling Language, p. 83-92, 1997.

CNN. CNN. Site da Internet da corporação que apresenta informações e notícias, inclusive informações meteorológicas. Disponível em: <http://www.cnn.com>. Acesso em: 10 de nov. 2000.

COAD, P. e YOURDON, E., **Object Oriented Analysis,** Prentice-Hall, s.l., 1990

COSMOPLAYER, **CosmoPlayer.** Site da Internet que disponibiliza o plug-in Cosmo para os principais navegadores da Internet, mantido pela CosmoSoftware Company. Disponível em: <http://www.cosmosoftware.com>. Acesso em: 15 de jan. 2000.

COSTA,. R. M. **Ambientes Virtuais na Reabilitação Cognitiva**. Rio de Janeiro, 2000. 134p. Tese (Doutorado). Engenharia de Sistemas e Computação - COPPE/UFRJ.

DEDE et al., Using virtual reality technology to convey abstract scientific concepts. In: M. J. JACOBSON & R. B. KOZMA (EDS.), LEARNING THE SCIENCES OF THE 21ST CENTURY: RESEARCH, DESIGN, AND IMPLEMENTING ADVANCED TECHNOLOGY LEARNING ENVIRONMENTS. Hillsdale, NJ: Lawrence Erlbaum.

Documento disponível na Web em <http://virtual.gmu.edu/SS_research/rpapers/jacobpdf.htm>. Acesso em Nov. 2000

DIONE. DIONE´S RESEARCH. Sítio com textos e pesquisas relativas a aplicação de RV na Educação e links para outras publicações. Disponível em: <http://express.sbs.ohio-state.edu/WEBCOURSE/ACKER/C659S96/ACKDOCS/EDUCSP96/EDUCDOCS/MURPHY/DIONESTF.HTM>. Acesso em: 10 de nov. 2000.

DOBSON, W. D.; RIESBECK, K. C. Tools for Incremental Development of Educational Interfaces. In: ACM Computer Human Interaction **- CHI´98**, USA, 1998. p. 384-391.

EAST CAROLINA UNIVERSITY. Virtual Reality And Education Laboratory. Greenville, North Carolina, USA.Contém diversas pesquisas sobre aplicações de RV na Educação. Disponível em: <http://eastnet.educ.ecu.edu/vr/>. Acesso em: 20 jan. 2000.

ELLIS, C. A; GIBBS, S. J.; REIN, G. L. Groupware: Some Issues and Experiences. **Communications of the ACM**, v.1. n. 34, , p. 38-58, Jan. 1991.

ELLIS R. S. What are Virtual Environments**. IEEE Computer Graphics and Applications**, p. 17-22, Jan. 1994.

ENCARNAÇÃO, J.; GOBEL, M. European Activities in Virtual Reality. **IEEE Computer Graphics and Applications**, p. 73, Jan. 1994.

FELDER, R. M.; SILVERMAN, L.K. Learning and Teaching Styles in Engineering Education. **Journal of Engineering Education**, 78(7), 674-681, Apr. 1988.

FOLEY et al. **Computer Graphics - Principles and Practice - Second Edition in C**. Addison Wesley Publishing Company. USA. July 1997.

FURLAN, J.D. **Modelagem de Objetos Através da UML -** ***The Unified Modeling Language*** Ed. Makron Books do Brasil, São Paulo, Brasil, 1998

GREEN. M; HALLIDAY, S., A Geometric Modeling and Animation System In: **Communications of the ACM**, v. 39, n. 5, p. 46-53, May 1996.

GROVE, A. Vá para a Internet ou morra. **Revista Info Exame**, Seção Clique Final, p. 99. out. 1999.

HILTZ, R. S.; WELLMAN, B. Asynchronous Learning Network as a Virtual Classroom In: **Communications of The ACM**, v. 40, n. 9, p. 44-49, Sept. 1997.

IEXPLORER 5.0, Microsoft Corporation. Página do Navegador para Internet da Microsoft. Disponível em: <http://www.microsoft.com.br/ie/>. Acesso em: 10 de nov. 2000.

IPOLITO, J.; KIRNER, C. Técnicas de Otimização e Realismo em Aplicações de Realidade Virtual usando VRML. In: Iº Workshop Brasileiro de Realidade Virtual, **WRV´97**, São Carlos, S.P, Brasil, 1997, p. 91-100.

IPOLITO, J.; KIRNER, C. Projeto de Ambientes Virtuais Multi-Usuários Usando Java e VRML. In: 3RD WORKSHOP ON VIRTUAL REALITY - **WRV 2000**, Gramado, RS, Brasil, out. 2000, p. 169 - 180.

JACOBSON, I. et al. **Object Oriented Software Engineering**. Addison Wesley Publishing Company, USA, 1992.

JAVA, *Java Devopment Kit - JDK.* Sítio da Internet que contém ferramentas para desenvolvimento de programas em Java mantido pela Sun Microsystens. Disponível em: <http://java.sun.com/>. Acesso em: 20 de nov. 2000.

JAVABRASIL. JavaBrasil. Sítio da Internet do grupo de desenvolvedores de Java no Brasil, que contém dicas da linguagem, lista de discussão, programas computacionais e chamadas para encontro do grupo. Disponível em: <http://www.soujava.org.br>. Acesso em 02 de nov. 2000.

JavaScript. **JavaScript Passo a Passo *Lite***, 1. Ed. Makron Books, São Paulo, 2001 (Série Lite)

JETS. - JETS Home Page. Java Enabled Tellecolaboration System, Version 2.02b. School o Information Technology and Engineering. University of Ottawa. Sítio do projeto JETS relacionado com a integração de Java com a Web, com soluções neste sentido. Disponível em: <http://www.mcrlab.uottawa.ca/jets>. Acesso em: 20 de nov. 1998.

JOHNSON C. Why ´Traditional´ HCI Techniques Fail to Support Desktop VR. In: The 3D Interface for the Information Worker - **IEE Coloquium**, 1998, p. 1/1 - 1/5.

KELLER, R.; SCHREIBER, J. GEO 3D: A Realidade Virtual como Suporte ao Ensino da Geometria Espacial. In: IIº WORKSHOP BRASILEIRO DE REALIDADE VIRTUAL - **WRV´99**. Marília, S.P., Brasil, 1999, p. 182 - 193.

KIRIL, J. et al. WebTOP: 3D Interactive Optics on the Web. In: ACM PROCEEDINGS OF THE WEB3D - VRML 2000 - FIFTH SYMPOSIUM ON THE VIRTUAL REALITY MODELING LANGUAGE. Monterey, CA - USA, Feb. 2000, p. 125-131.

KIRNER, C. Sistemas de Realidade Virtual. Apostila publicada pelo Departamento de Computação da Universidade Federal de São Carlos. São Carlos - SP, Brasil, 1997.

KIRNER, T.; MARTINS, V. A Model of Software Development Process for Virtual Environments: Definition and a case Study. In: 2^{O} IEEE INTERNATIONAL SYMPOSIUM ON APPLICATION-SPECIFIC SYSTEMS AND SOFTWARE ENGINEERING AND TECHNOLOGY – **ASSET'98**, Richardson, USA, 1998.

KRUEGER, M.W., **Artificial Reality II**, Addison Wesley, Reading, MA, 1991

KUBO, M. M. **Suporte de Comunicação para Sistemas Colaborativos Interoperáveis de Realidade Virtual**. São Carlos (S.P.), 2000. 192p. Dissertação (Mestrado). Universidade Federal de São Carlos - UFSCar.

KUBO, M. M.; DERRIGI, F.; KIRNER, C. Interação nos Ambientes Virtuais Colaborativos de Ensino. In: 3RD WORKSHOP ON VIRTUAL REALITY - **WRV 2000**, Gramado, RS, Brasil, out. 2000, p. 55 - 64.

LABVIRT. LABORATÓRIO VIRTUAL DE FÍSICA. Sítio de animações de Física do Instituto de Física da Universidade de São Paulo da USP. Disponível em: <http://www.labvirt.if.usp.br>. Acesso em: nov. 2000.

LATTA J.N.; OBERG D.J. A Conceptual Virtual Reality Model. **IEEE Computer Graphics and Applications**. Jan. 1994, p. 23-28.

LAWHEAD, B. P. et al. The Web and distance learning: what is appropiate and what is not., In: ACM Procedings of the conference on Integrating Technology into Computer Science Education - **ITiCSE´97**, 1997, p. 27-37.

LAWTON, M, Advancing 3D trough VRML on the Web. **IEEE Computer Graphics and Applications**, v. 19, n. 2, p. 4 - 5, Mar./Apr. 1999.

LEE, J.H.; PRAKASH, A. Supporting Multi-User, Multi-Applet Workspaces in CBE. **Proc. CSCW´96**, v. 1, p. 344-353, 1996.

LEWIS, C.; GRIFFIN, M. Human Factors Consideration in Clinical Applications of Virtual Reality. **Virtual Reality in Neuro-Psycho-Physiology**. Ed. Giuseppe Riva, IOS Press, 1997.

LEWIS, C. et al. Adapting User Interface Desing Methods to the Design of Educational Activities. In: **ACM CHI´98**, Los Angeles, CA, USA, 1998, p. 619- 626.

LICKLIDER, J.C.R e TAYLOR, R.W. The Computer as Communication Device, **Sci. Tech**, p. 21-31, Apr. 1968.

MAÇADA, L.D.; BASSO, A.V.M. Mathematikos: Educação e Treinamento à Distância,. In: Simpósio Brasileiro de Informática na Educação - **SBIE 2000**, Maceió, AL, nov. 2000, p. 199-204.

MARRIN C. External Authoring Interface Working Group. Sítio da Internet sobre a interface de Java com VRML, via EAI. Contém dicas e instruções sobre a forma de integração das duas linguagens, dicas e notas, além de exemplos. Disponível em: <http://www.web3d.org/WorkingGroups/vrml-eai/>. Acesso em: abr. 2001.

MARTINS, F. V; KUBO, M. M.; KIRNER, T. Análise, Projeto e Implementação de uma Aplicação de Realidade Virtual Cooperativa. In: II WORKSHOP BRASILEIRO DE REALIDADE VIRTUAL, **WRV´99**, Marília , São Paulo, Brasil, 1999, p. 43 - 55.

MATSUBA, N.; STEPHEN; ROEHL, B. Bottom, Thou Art Translated: The Making of VRML Dream. **IEEE Computer Graphics and Applications**, v. 19, n. 2, p. 45-51, Mar./Apr. 1999.

MEC – Ministério da Educação e Cultura. Sítio da Internet sobre Educação a Distância. Contém cursos regulamentados, aspectos jurídicos e normativos do Ensino a Distância no Brasil, além de instituições credenciadas, relatórios etc. Disponível em <http://www.mec.gov.br/seed/tvescola/regulamentacaoEAD.shtm>. Acesso em 06 nov. 2001

MEIGUINS, S. B.; BEHRENS, H. F. Laboratório Virtual para Experiências de Eletrônica. In: II WORKSHOP BRASILEIRO DE REALIDADE VIRTUAL, **WRV´99**, Marília , São Paulo, Brasil, 1999, p. 56-67.

MYERS, A. B. A Brief History of Human Computer Interaction Technology. **ACM Interaction**, v. 5, n. 2, Mar. 1998.

NADEAU, D. R. Building Virtual Worlds with VRML. **IEEE Computer Graphics and Apllications**, v.19, n.2, p.19-29, Mar./Apr. 1999.

NISKIER, A. **Educação a Distância, a Tecnologia da Esperança.** Editora Loyola, Rio de Janeiro, 1999.

OST, C. L.; INDRUSIAK, S. LEANDRO; R. Dynamic 3D Models of Integrated Circuits using VRML. In: II WORKSHOP BRASILEIRO DE REALIDADE VIRTUAL, **WRV´99**, Marília , São Paulo, Brasil, 1999, p. 95-101.

PANTELIDIS V. Vesamontex. Sítio do projeto e descrição detalhada das atividades e resultados da implementação de uma solução de VR aplicada a Educação. Disponível em: <http://users.hub.ofthe.net/~mtalkmit/veshtml2.html>. Acesso em: nov. 1999.

PAUSCH, R.; PROFFIT, D.; WILLIAMS, G. Quantifying Immersion in Virtual Reality In: Proceedings of the 1997 - ACM Siggraph annual conference on Computer Graphics, 1997, p. 13-18.

PHYSICSWEB. PhysicsWeb - resources. Sítio que reúne animações em Física utilizando VRML e Shockwave, na forma de um laboratório virtual de Física. Disponível em: <http://www.physicsweb.org/TIPTOP/VLAB>. Acesso em: 05 de nov. 2000.

PIERCE, S. J. et al. Alice: Easy to Use Interactive 3D Graphics. In: Proceedings of the 10th annual ACM Symposium on User Interface Software and Technology, 1997, p. 77.

PINHO, M. VII Simpósio Brasileiro de Informática na Educação. Documento sobre a utilização de Realidade Virtual como Ferramenta na Educação, que foi apresentado como Tutorial do Simpósio Brasileiro de Informática na Educação, SBIE´96, Belo Horizonte, 1996. Disponível em: <http://grv.inf.pucrs.br/Pagina/Educa/educa.htm>. Acesso em: set. 1999.

PINHO, M. Interação em Ambientes Tridimensionais. Tutorial do 3rd WORKSHOP ON VIRTUAL REALITY - **WRV´2000**, Gramado, RS, Outubro/2000.

PINHO, M. Grupo de Realidade Virtual da Pontíficia Universidade Católica do Rio Grande do Sul - PUC/RS. Disponível em <http://grv.inf.pucrs.br/>. Acesso em: 10 de nov. 2000.

PRESSMAN, R. **Engenharia de Software**, Trad. de José Carlos Barbosa dos Santos, São Paulo, Makron Books, 1995.

PSS - SCIENCESPACE PROJECT/ Virtual Reality at GMU Homepage. University of Houston. Sítio com a documentação e implementações de experimentos disponíveis em VRML. Disponível em: <http://www.virtual.gmu.edu/vrhome.htm>. Acesso em: 10 de jul. 1999. Sítio disponível desde 1997.

PYTHON. Sítio relativo à linguagem Phyton. Disponível em <http://www.python.org/>. Acesso em: nov. 2000.

RAPOSO, A.; MAGALHÃES, P. L. R.; L. M. I. Interação na WEB. In: XVIII JAI - JORNADA DE ATUALIZAÇÃO EM INFORMÁTICA (JAI'99) do XIX CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO - SBC'99, Rio de Janeiro, RJ, 1999. Documento disponível em: <http://www.dca.fee.unicamp.br/~alberto/pubs/JAI99/curso_jai99.html>. Acesso em: 20 ago. 1999.

RAPOSO, A. Interação na WEB - Tendências. In: SEMINÁRIO INTERNO DE TÓPICOS DE ENGENHARIA DE COMPUTAÇÃO DA UNICAMP. Campinas, S.P. 1999.

REDDY, M.; LECLERC, Y.; IVERSON, L e BLETTER, N. Terra Vision II: Visualizing Massive Terrain Databases in VRML. **IEEE Computer Graphics and Applications**, v. 19, n. 2, Mar./Apr. 1999.

RITHCEY T. **Programando com Java! Beta 2.0 - 1ª Edição**. Editora Campus, Rio de Janeiro, 1997. 425 p.

ROBLES, T. et al. Using Multimedia Communication Technologies in Distance Learning. In: ACM PROCEDINGS OF THE CONFERENCE ON INTEGRATING TECHNOLOGY INTOCOMPUTER SCIENCE EDUCATION - **ITiCSE´97**, USA, 1997, p. 06-07.

RUDOMIN, I. Arquitectura para realidad virtual distribuída com física simulada, problada de agentes autónomos. In: **Computacion Visual 1997**. Disponível em: <http://sgio2.cem.itesm.mx/rudomim/ >. Acesso em: fev. 1999.

SABA, F. Introduction to Distance Education. Distance-Educator.com. Sítio que contém diversos artigos sobre Educação a Distância. Disponível em: <http://www.distance-educator.com>. Acesso em: dez.1999.

SALGADO, R. VRML Gallery of Eletromagnetism (by Rob Salgado). Sítio da Internet que contém animações em VRML relacionadas com Física, tais como a Lei de Ámpere, campos elétricos de cargas puntiformes, dipolos elétricos etc. Disponível em: <http://www.physics.syr.edu/courses/vrml/electromagnetism/>. Acesso em: 20 de jan. 2001.

SALZMAN, M. et al. Learner Centered Design of Sensorily Immersive Microworlds Using a Virtual Reality Interface. In: PROCEEDINGS OF THE SEVENTH INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND EDUCATION**,** Charlottesvillle, VA, 1995, p. 554-564.

SANCHES, C.I; PRADO, F. A. Framework para Ensino a Distância via Web - Ferramentas para Internet/Intranet/WWW, In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO - **SBIE 2000**, Maceió, AL, nov. 2000, p.183-190.

SCHANK, R.C. Revolutionizing the Tratidional Classroom Course. **Communications of ACM,** v. 4, n. 12., p.21-24, Dec. 2001

SCOTT, A. The Marriage of Java and VRML. VRML Site Magazine. Sítio que disponibiliza informações importantes sobre a união das linguagens Java e VRML. Disponível em: <http://www.vrmlsite.com>. Acesso em: 10 de abr. 2000.

SHERMAN, R. W. et al. Experiences with Virtual Reality applications. In: PROCEEDINGS OF THE 24TH ANNUAL CONFERENCE ON COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES - **SIGGRAPH '97**. USA, 1997, p. 473-476.

SHNEIDERMAN, B., **Designing the User Interface: Strategies for Effective Human-Computer Interaction - 2nd Edition**, Ed. Addison Wesley, USA, 1992.

SILVA, C.A. J. Uso de Gabaritos Configuráveis para Desenvolvimento de Interfaces Virtuais. In: Iº WORKSHOP BRASILEIRO DE REALIDADE VIRTUAL - **WRV´97**, São Carlos, S.P., 1997, p. 81-90.

SILVA, T. A.; OSÓRIO, F. Modelagem de objetos 3D em VRML: Uma implementação multiplataforma orientada ao ensino. In: 3RD WORKSHOP ON VIRTUAL REALITY - **WRV 2000**, Gramado, RS, Brasil, out. 2000, p. 16-18.

SILVEIRA, F. I; PELLEGRINO, M.R. S. Modelagem de Sólidos Aplicada à Construção de Mundos Virtuais. In: Iº Workshop Brasileiro de Realidade Virtual - **WRV´97**, São Carlos, S.P., 1997, p. 52-61.

SOUZA, L. A. ; OLIVEIRA, J. C, "Recursos da computação gráfica para o Desenvolvimento de um Laboratório Virtual de Teoria Eletromagnética. In: XXIX CONGRESSO BRASILEIRO DE ENSINO DE ENGENHARIA. Porto Alegre, Set. 2001

TERRA, TERRA - Portal da Internet brasileiro. O Sítio contém uma área específica para educação, onde podem ser encontradas simulações em VRML relativas a Física e alguns conteúdos interessantes na forma de hipertextos. Disponível em: <http://www.terra.com>. Acesso em: 12 nov. 2000.

TORI, R.; FERREIRA, M. A. G. V., Educação sem a distância em Cursos de Informática. In: XV Congresso da Sociedade Brasileira de Computação - VII Workshop sobre Educação em Informática. Rio de Janeiro, 1999, p. 581-590

TORI, R., "Maps and Spatial Metaphors in Hypermedia Systems", **Revista Graf&Tec**, v. 0, n. 0., p. 27-38, jul. 1996.

VIRTUAL LABORATORY - O sítio disponibiliza uma série de *applet* Java com simulações de diversos experimentos de Física, Química e Circuitos Elétricos. Disponível em: <http://www.jhu.edu/~virtlab/virtlab.html>. Acesso em: 03 maio 2002.

VLAB - Oregon - O sítio disponibiliza um conjunto de experimentos virtuais interativos concebidos em Java (*applets*). Disponível em <http://jersey.uoregon.edu/vlab/>. Acesso em 02 fevereiro 2002

VRMLPAD. Parallellgraphis. O sítio disponibiliza diversos programas computacionais de grande utilidade para desenvolvimento de ambientes virtuais em VRML. Disponível em: <http://www.parallellgraphics.com>. Acesso em: 02 nov. 2000.

WINTERS, P. et al. **Aprenda em 21 dias Visual J++**; tradução de João E. N. Tortello, Rio de Janeiro: Campus, 1997. 662 p.

ZELEZNIK, R. C.; MARKOSIAN, L.; LAVIOLA, J.J. e FORSBERG, A. S. Seamless Interaction in Virtual Reality. **IEEE Computer Graphics and Applications**, p. 6-9, Nov./Dec. 1997.